

1-1-2011

# Complete and Exact Small Signal Analysis of DC-to-DC Switched Power Converters Under Various Operating Modes and Control Methods

Justin French Simmons  
*Portland State University*

**Let us know how access to this document benefits you.**

Follow this and additional works at: [http://pdxscholar.library.pdx.edu/open\\_access\\_etds](http://pdxscholar.library.pdx.edu/open_access_etds)

---

## Recommended Citation

Simmons, Justin French, "Complete and Exact Small Signal Analysis of DC-to-DC Switched Power Converters Under Various Operating Modes and Control Methods" (2011). *Dissertations and Theses*. Paper 195.

[10.15760/etd.195](https://pdxscholar.library.pdx.edu/open_access_etds/10.15760/etd.195)

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. For more information, please contact [pdxscholar@pdx.edu](mailto:pdxscholar@pdx.edu).

Complete and Exact Small Signal Analysis of DC-to-DC Switched Power  
Converters Under Various Operating Modes and Control Methods

by

Justin French Simmons

A thesis submitted in partial fulfillment of the  
requirements for the degree of

Master of Science  
in  
Electrical & Computer Engineering

Thesis Committee:  
Richard Tymerski, Chair  
Garrison Greenwood  
Fu Li

Portland State University  
©2011

## Abstract

A method to obtain the exact control-to-output and input-to-output transfer functions for switched DC-to-DC pulse-width modulated power converters is applied to different combinations of operating conditions such as continuous conduction mode (CCM), discontinuous conduction mode (DCM), and discontinuous voltage mode (DVM) and methods of control such as normal voltage mode (NVM), current programming mode (CPM), and  $V^2$  control. The majority of these combinations have not previously had their transfer functions of interest derived to the accuracy provided by the method. The derivation of the method is covered, and the results from its application are verified by simulation. The simulation method is also justified and proposed as an improvement to the methods used by simulation engines for switched mode power supply design such as SIMPLIS.

## Table of Contents

|  |     |
|--|-----|
| Abstract . . . . .   | i   |
| List of Figures . . . . .  | iii |
| Ch. 1: Introduction . . . . .  | 1   |
| Ch. 2: Large and Small Signal State Space Model . . . . .            | 9   |
| Ch. 3: The Exact Small Signal Transfer Function Expression . . . . . | 17  |
| Ch. 4: Normal Voltage Mode Transfer Functions . . . . .              | 23  |
| Ch. 5: Current Programmed Transfer Functions . . . . .               | 50  |
| Ch. 6: $V^2$ Controlled Transfer Functions . . . . .                 | 79  |
| Ch. 7: Verification . . . . .  | 108 |
| Conclusion . . . . .   | 112 |
| References . . . . .   | 114 |
| Appendix: MATLAB Code . . . . .                                      | 115 |



## List of Figures

|  |    |
|--|----|
| 1. General DC-to-DC Power Converter . . . . .                  | 2  |
| 2. Square Wave Regulating Topologies . . . . .                 | 2  |
| 3. Boost Converter Power Stage . . . . .                       | 5  |
| 4. OTB Converter Power Stage . . . . .                         | 5  |
| 5. Boost Converter's Effective Topologies . . . . .            | 8  |
| 6. OTB Converter's Effective Topologies . . . . .              | 8  |
| 7. Period $T_x$ with Perturbations . . . . .                   | 22 |
| 8. NVM Boost Converter . . . . .                               | 24 |
| 9. NVM OTB Converter . . . . .                                 | 25 |
| 10. $T_s$ under Model 1 and Model 2 . . . . .                  | 27 |
| 11. Effective Topologies of a NVM Boost Converter in CCM . . . | 27 |
| 12. NVM $T_s$ in CCM . . . . .                                 | 28 |
| 13. NVM Converter in CCM Control-to-Output . . . . .           | 32 |
| 14. NVM Converter in CCM Input-to-Output . . . . .             | 32 |
| 15. Effective Topologies of a NVM Boost Converter in DCM . . . | 34 |
| 16. NVM $T_s$ in DCM . . . . .                                 | 35 |
| 17. NVM Converter in DCM Control-to-Output . . . . .           | 40 |
| 18. NVM Converter in DCM Input-to-Output . . . . .             | 40 |
| 19. Effective Topologies of a NVM OTB Converter in DVM . . .   | 42 |
| 20. NVM $T_s$ in DVM . . . . .                                 | 43 |
| 21. NVM Converter in DVM Control-to-Output . . . . .           | 49 |
| 22. NVM Converter in DVM Input-to-Output . . . . .             | 49 |
| 23. CPM Boost Converter . . . . .                              | 51 |
| 24. CPM OTB Converter . . . . .                                | 52 |
| 25. Effective Topologies of a CPM Boost Converter in CCM . . . | 54 |

|     |   |     |
|-----|---|-----|
| 26. | CPM Stability Ramp for $D > 0.5$ . . . . .                        | 54  |
| 27. | CPM $T_s$ in CCM . . . . .  | 55  |
| 28. | CPM Converter in CCM Control-to-Output . . . . .                  | 60  |
| 29. | CPM Converter in CCM Input-to-Output . . . . .                    | 60  |
| 30. | CPM $T_s$ in DCM . . . . .  | 62  |
| 31. | Effective Topologies of a CPM Boost Converter in DCM . . . .      | 63  |
| 32. | CPM Converter in DCM Control-to-Output . . . . .                  | 69  |
| 33. | CPM Converter in DCM Input-to-Output . . . . .                    | 69  |
| 34. | CPM $T_s$ in DVM . . . . .  | 71  |
| 35. | Effective Topologies of a CPM OTB Converter in DVM . . . .        | 72  |
| 36. | CPM Converter in DVM Control-to-Output . . . . .                  | 78  |
| 37. | CPM Converter in DVM Input-to-Output . . . . .                    | 78  |
| 38. | $V^2$ Controlled Boost Converter . . . . .                        | 80  |
| 39. | $V^2$ Controlled OTB Converter . . . . .                          | 81  |
| 40. | Effective Topologies of a $V^2$ controlled Boost Converter in CCM | 83  |
| 41. | $V^2$ Control Stability Ramp for $D > 0.5$ . . . . .              | 83  |
| 42. | $V^2$ Control $T_s$ in CCM . . . . .                              | 84  |
| 43. | $V^2$ Controlled Converter in CCM Control-to-Output . . . . .     | 89  |
| 44. | $V^2$ Controlled in CCM Input-to-Output . . . . .                 | 89  |
| 45. | $V^2$ Control $T_s$ in DCM . . . . .                              | 91  |
| 46. | Effective Topologies of a $V^2$ Controlled Boost Converter in DCM | 92  |
| 47. | $V^2$ Controlled Converter in DCM Control-to-Output . . . . .     | 98  |
| 48. | $V^2$ Controlled Converter in DCM Input-to-Output . . . . .       | 98  |
| 49. | $V^2$ Control $T_s$ in DVM . . . . .                              | 100 |
| 50. | Effective Topologies of a $V^2$ OTB Converter in DVM . . . . .    | 101 |
| 51. | $V^2$ Controlled Converter in DVM Control-to-Output . . . . .     | 107 |
| 52. | $V^2$ Controlled Converter in DVM Input-to-Output . . . . .       | 107 |

|     |   |     |
|-----|---|-----|
| 53. | Approximation of Sinusoidal Perturbation Source . . . . . | 110 |
|-----|---|-----|

## Chapter 1: Introduction

Switching power converters allow efficient interfacing of a power supply to subcircuits which require a different operating voltage, transferring power by switching between topologies at a rate determined by a duty ratio  $d$  which can be adjusted to reliably maintain an average (DC) output voltage  $V_{out}(t)$  despite variations of the supply voltage  $v_g(t)$ . The power stages of switched converters consist of only storage elements and switches making the power conversion theoretically lossless with ideal components. They are essential for portable devices whose supply voltage will lower as batteries are drained and for integrated circuits whose integrity would be severely compromised by heat dissipation. Fig. 1 shows a representation of a DC-to-DC power converter in normal voltage mode (NVM). For this thesis, NVM is used to describe a converter whose sawtooth signal is externally supplied rather than generated by some output of the system. All converters in this thesis will feature an output voltage controller which serves to adjust  $d$  in order to counteract variations in  $v_g(t)$  and consists of a frequency compensator to ensure a positive phase margin. In Fig. 1, the output voltage  $v_{out}(t)$  across the load resistance  $R_l$  is sensed through a gain  $H(s)$  and fed into the output voltage controller along with an external reference voltage  $V_{ref}$ . Impedances  $Z_1$  and  $Z_2$  determine the gain and the pole and zero locations and are made up of capacitors, resistors, or both depending on the particular compensator used in the output voltage controller. The compensator produces a control input signal  $r(t)$ :

$$r(t) = V_{ref} + \frac{Z_2}{Z_1}[V_{ref} - v_{out}(t)H(s)] \quad (1.1)$$

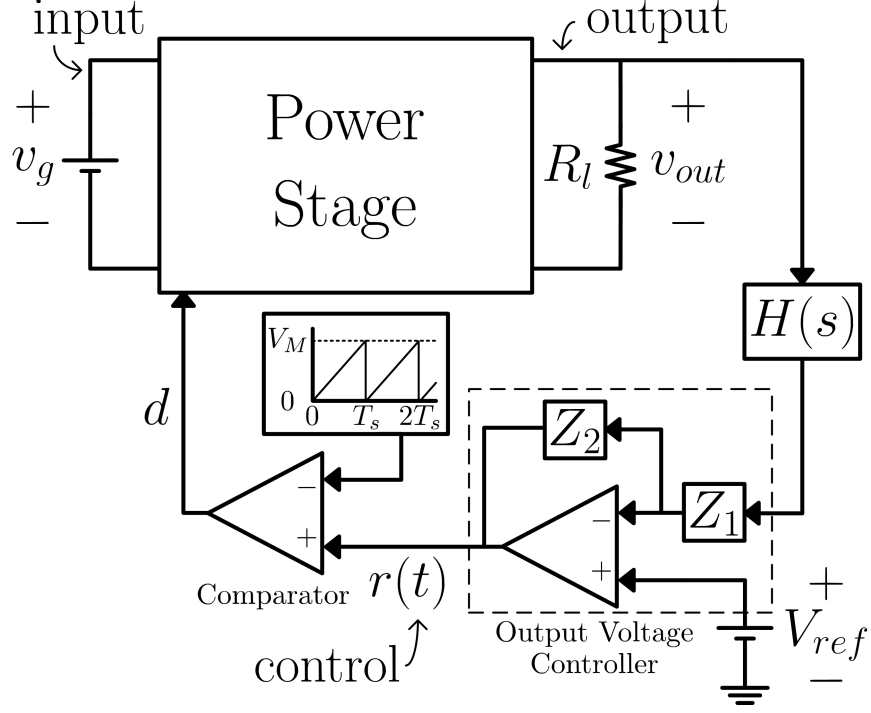


Figure 1: General diagram of an NVM DC-to-DC power converter featuring an output voltage controller. The output voltage is sensed through  $H(s)$  and fed through a compensator to produce a signal  $r(t)$ , which is then used by a comparator along with an externally provided sawtooth signal to produce a square wave with period  $T_s$  and duty ratio  $d$  that controls the active switch. The input  $v_g(t)$  contains both DC ( $V_g$ ) and AC ( $\hat{v}_g(t)$ ) components.

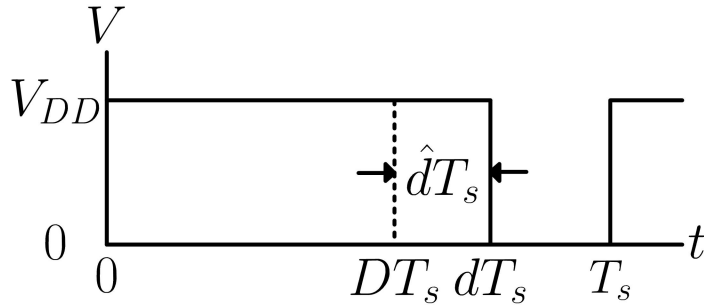


Figure 2: Square wave produced by the comparator which regulates how long the active switch  $Q$  is turned on. The time  $dT_s$  is the actual time that the active switch is on in the presence of perturbations while  $DT_s$  is the DC average time that the active switch is on in the absence of perturbations.

which is then fed to a comparator that compares it to an externally introduced sawtooth signal of period  $T_s$  and amplitude  $V_M$ , producing a square wave with a duty ratio  $d$  and switching period  $T_s$  as illustrated in Fig. 2 that regulates the switching between topologies of the power stage. Adjusting  $H(s)$  appropriately so that  $V_{ref} = V_{out}(t)H(s)$ , where  $V_{out}(t)$  is the DC component of  $v_{out}(t)$ , makes:

$$R = V_{ref} \quad (1.2)$$

where  $R$  is the constant large signal (DC) component of  $r(t)$ .  $D$ , the DC component of  $d$ , is then determined by:

$$D = \frac{V_{ref}}{V_M} \quad (1.3)$$

A good power converter will have minimal change in  $v_{out}(t)$  with respect to change in  $v_g(t)$ , thus designing a power converter requires the satisfaction of a minimizing sensitivity of the output voltage to variations in the input voltage, or audiosusceptibility, represented by  $\frac{\hat{v}_{out}}{\hat{v}_g}$  over a specified frequency range where  $\hat{v}_g(t)$  and  $\hat{v}_{out}(t)$  are the small signal (AC) components of the input  $v_g(t)$  and  $v_{out}(t)$ , respectively. The design of a compensator requires the control-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{r}}$  where  $\hat{r}$  is the AC component of the control input signal  $r(t)$ . This thesis focuses on the development of an already derived method to determine these transfer functions exactly [2,3] and applying this exact method to different combinations of operating modes and control methods, some of which have not yet had their exact transfer functions derived. The method is exact in the "small signal sense" as higher order terms are dropped. This is the only approximation made by the method. The control-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{r}}$  is found for the open loop case (no

output voltage controller, no outer loop), where  $r(t)$  is externally supplied, which is essential for the design of the output voltage controller.

The next chapter will serve to develop a mathematical representation of the system using state-space analysis, starting with the DC model and perturbing the expressions to obtain the AC model. A general expression  $H_0(j\omega)$  will then be developed by the proper application of the time-varying function as described in [1], from which  $\frac{\hat{v}_{out}}{\hat{r}}$  and  $\frac{\hat{v}_{out}}{\hat{v}_g}$  will be isolated. While [2] and [3] have outlined the procedures to develop  $\frac{\hat{v}_{out}}{\hat{r}}$  and  $\frac{\hat{v}_{out}}{\hat{v}_g}$  separately, this thesis will develop both transfer functions together and extract them afterwards. From the application and verification of the exact method to different combinations of operating modes and control methods it will become evident that the method is widely applicable and exact in the "small signal sense," as well as amenable to computer implementation and automation. Having the method that produces exact results also has the potential for developing simplified models for widespread use with minimal loss of accuracy.

This method will be applied to a well known converter operating in continuous conduction mode (CCM) and discontinuous conduction mode (DCM), first in NVM, whose inner loop is open, and then in current programmed mode (CPM) and  $V^2$  control. CPM and  $V^2$  control both generate a sawtooth signal from an output of the system which leads to advantages in converter performance. The well known converter that will be used for these examples will be a boost converter, the power stage of which is shown in Fig. 3 and whose steady state DC conversion ratio is  $\frac{V_{out}}{V_g} = \frac{1}{1-D}$ .

The method will also be applied to the optimum topology buck (OTB) converter, whose power stage is shown in Fig. 4 and whose steady state DC conversion ratio is  $\frac{V_{out}}{V_g} = D$ , for operation in discontinuous voltage mode

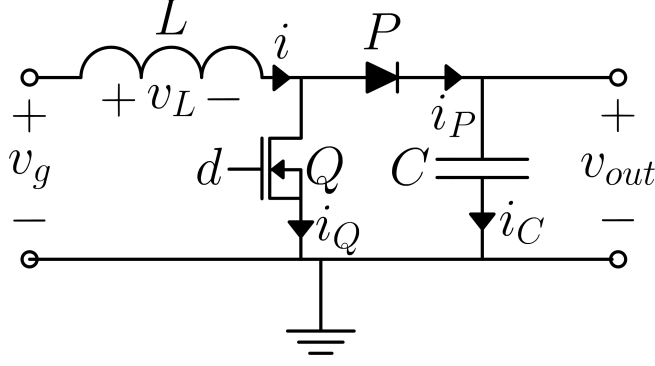


Figure 3: The power stage of the well-known boost converter, consisting of one capacitor  $C$ , one inductor  $L$ , a transistor  $Q$  and a switching diode  $P$ . A square wave with duty ratio  $d$  and period  $T_s$  produced by the comparator controls the operation of  $Q$ , also called the active switch, which in turn controls the state of  $P$ , also called the passive switch.

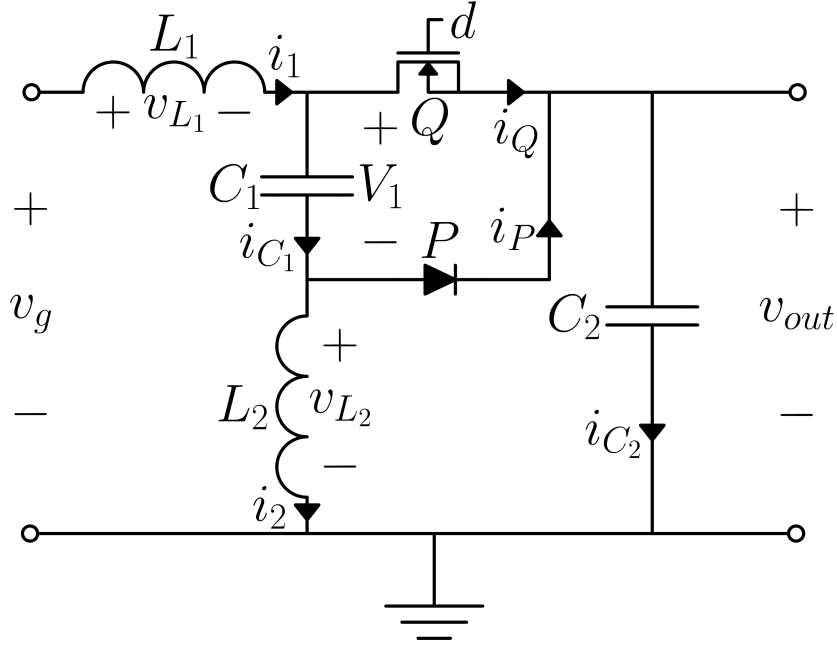


Figure 4: The power stage of the optimum topology buck (OTB) converter, consisting of two capacitors  $C_1$  and  $C_2$ , two inductors  $L_1$  and  $L_2$ , a transistor  $Q$  and a switching diode  $P$ . Capacitor  $C_1$  is also called the storage capacitor which will turn on  $P$  prematurely if its voltage  $v_1(t)$  drops to zero when operating in DVM. A square wave with duty ratio  $d$  and period  $T_s$  produced by the comparator controls the operation of  $Q$ , also called the active switch, which in turn controls the state of  $P$ , also called the passive switch.



(DVM) under the previously mentioned methods of control. The following component values for the power stages of the boost and OTB converters will be used:

Boost Converter:

$$\begin{aligned} L &= 58\mu H \\ C &= 5.5\mu F \end{aligned} \quad (1.4a)$$

OTB Converter:

$$\begin{aligned} L_1 &= 130\mu F \\ L_2 &= 1mH \\ C_1 &= 60nF \\ C_2 &= 50\mu F \end{aligned} \quad (1.4b)$$

For all of the operating modes and control methods investigated in this thesis, the square wave produced by the comparator controls the active switch, realized in this thesis to be a transistor  $Q$ . When the square wave is high,  $Q$  is on which puts a negative voltage across the switching diode  $P$  (or the passive switch) and shuts it off, defining the first subinterval. When the square wave goes to zero,  $Q$  turns off and  $P$  turns on defining the second subinterval. For a converter operating in CCM, these are the only two subintervals that exist within a switching cycle.

A converter operates in DCM when the current in  $P$  drops to zero and  $P$  shuts off, introducing a third subinterval in which both switches are off. DVM operation occurs when the voltage across a storage capacitor (capacitor  $C_1$  in the OTB converter, which is sized appropriately in (1.4b)) drops to zero and causes  $P$  to turn on prematurely, introducing a subinterval in between the first and second subintervals where both switches are on. Both DCM and DVM operation make the conversion ratio load dependent. It should be noted that because the boost converter has no storage capacitor, DVM is not applicable to it. For this thesis, it will be assumed that the operating conditions CCM,

DCM, and DVM are all mutually exclusive cases.

Fig. 5 and Fig. 6 show the possible effective topologies of the power stages of the boost and OTB converters, respectively, assuming that  $Q$  and  $P$  are ideal switches. Ideal switches are considered to have a resistance of  $0\Omega$ , or a wire, when turned on and a resistance of  $\infty\Omega$ , or an open circuit, when turned off. This thesis also assumes copper loss and other parasitics to be negligible. State equations can be derived from these effective topologies which will then be used in the state space models explained in the next chapter. The state space models will serve as a mathematical representation of the converter from which a general expression containing both  $\frac{v_{out}}{\hat{r}}$  and  $\frac{v_{out}}{\hat{u}}$  can be derived.

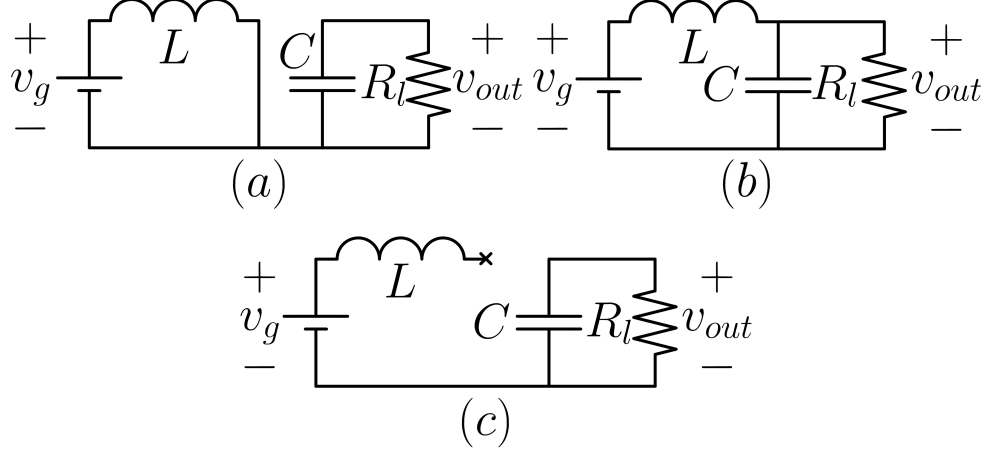


Figure 5: The effective topologies of the boost converter assuming ideal switching elements. (a) applies to the first subinterval of CCM operation, (b) applies to the second subinterval of CCM operation, (c) applies to the subinterval in DCM operation where both switches are off.

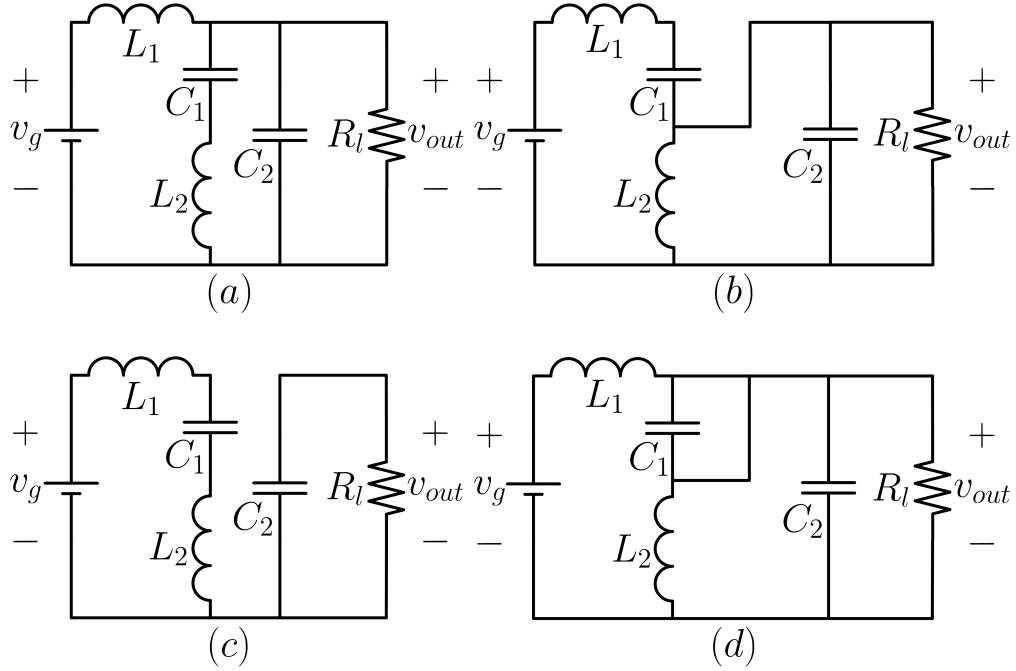


Figure 6: The effective topologies of the OTB converter assuming ideal switching elements. (a) applies to the first subinterval of CCM operation, (b) applies to the second subinterval of CCM operation, (c) applies to the subinterval in DCM operation where both switches are off and (d) applies to the subinterval in DVM operation where both switches are on.

## Chapter 2: Large and Small Signal State Space Model

State space analysis will be utilized to develop expressions that describe the states of the switching converters. The large signal DC model will first be described and then perturbed to obtain the small signal AC model. While there is generally one switching device that controls the switching between two topologies, more topologies may exist should certain conditions be satisfied such as in DCM or DVM. An integer  $N_s$  denotes the number of topologies that are present within a switching period  $T_s$ . The time spent in a particular topology will define a subinterval so that  $T_s$  is divided into  $N_s$  different subintervals. The general state space representation for a time varying system is:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (2.1a)$$

$$y(t) = C^T(t)x(t) + E(t)u(t) \quad (2.1b)$$

where  $x(t)$  is the state vector consisting of both the large signal and small signal values of each inductor current and capacitor voltage,  $\dot{x}(t)$  is its derivative,  $y(t)$  is the output of the system, and  $u(t)$  is the input vector.  $v_g(t)$  is the only input considered in this thesis, making  $u(t)$  a scalar:

$$x(t) = \begin{bmatrix} i_1(t) \\ \vdots \\ i_k(t) \\ v_1(t) \\ \vdots \\ v_l(t) \end{bmatrix} = \begin{bmatrix} I_1(t) + \hat{i}_1(t) \\ \vdots \\ I_k(t) + \hat{i}_k(t) \\ V_1(t) + \hat{v}_1(t) \\ \vdots \\ V_l(t) + \hat{v}_l(t) \end{bmatrix} \quad (2.2)$$

$$\begin{aligned} u(t) &= U + \hat{u}(t) \\ &= v_g(t) \\ &= V_g + \hat{v}_g(t) \end{aligned} \quad (2.3)$$

for  $k$  inductors and  $l$  capacitors in the power stage of the converter. For a piecewise linear system the equations in (2.1) can be replaced by a set of  $N_s$  linear time invariant (LTI) state equations of the form:

$$\dot{x}(t) = A_i x(t) + B_i u(t) \quad (2.4a)$$

$$y(t) = C_i^T x(t) + E_i u(t) \quad (2.4b)$$

for  $t_i \leq t \leq t_{i+1}$ ,  $i \in (1, 2, \dots, N_s)$ , where  $A_i$  and  $B_i$  are the state and input matrices, respectively, that find the state derivative  $\dot{x}(t)$  from  $x(t)$  and  $u(t)$  for the  $i$ th subinterval of  $T_s$ .  $C_i^T$  and  $E_i$  are the output and feedthrough matrices, respectively, that extract a desired output  $y(t)$  from  $x(t)$  and  $u(t)$  for the  $i$ th subinterval of  $T_s$ . Solving the state equation (2.4a) yields:

$$x(t) = e^{A_i(t-t_i)} x(t_i) + \int_{t_i}^t e^{A_i(t-\tau)} B_i u(\tau) d\tau \quad (2.5)$$

for  $t_i \leq t \leq t_{i+1}$ , from which the value at the end of the interval can be expressed:

$$x(t_{i+1}) = e^{A_i(t_{i+1}-t_i)} x(t_i) + \int_{t_i}^{t_{i+1}} e^{A_i(t_{i+1}-\tau)} B_i u(\tau) d\tau \quad (2.6)$$

where  $t_i = T_i + \hat{t}_i$ .  $T_i$  is the DC component of  $t_i$  and  $\hat{t}_i$  is the AC component which is nonzero when a perturbation is introduced to the system.

For large signal analysis no perturbations are considered making  $t_i = T_i$ ,  $x(t) = X(t)$ , and  $u(t) = U$ . The large signal form of (2.4a) is:

$$\dot{X}(t) = A_i X(t) + B_i U \quad (2.7a)$$

$$Y(t) = C_i^T X(t) + E_i U \quad (2.7b)$$

and the large signal form of (2.6) is:

$$X(T_{i+1}) = e^{A_i(T_{i+1}-T_i)}X(T_i) + \int_{T_i}^{T_{i+1}} e^{A_i(T_{i+1}-\tau)}B_iU d\tau \quad (2.8)$$

Evaluating the integral results in:

$$X(T_{i+1}) = e^{A_i(T_{i+1}-T_i)}X(T_i) + A_i^{-1}[e^{A_i(T_{i+1}-T_i)} - I]B_iU \quad (2.9)$$

Letting  $\Phi_i = e^{A_i(T_{i+1}-T_i)}$  and  $\Psi_i = A_i^{-1}[\Phi_i - I]B_i$ , (2.9) becomes:

$$X(T_{i+1}) = \Phi_i X(T_i) + \Psi_i U \quad (2.10)$$

for  $i \in (1, 2, \dots, N_s)$ , which is the large signal difference equation. Equation (2.10) will be solved for a finite  $N_s$  as it will be required for later use. The cases covered in this thesis are only of two and three topologies so (2.10) will only be solved for  $N_s = 2$  and  $N_s = 3$ . Doing so makes use of the following condition for a converter in large signal steady state:

$$X(t) = X(t + nT_s) \quad (2.11)$$

for any nonnegative integer  $n$ .

For  $N_s = 2$ , (2.10) becomes:

$$X(T_2) = \Phi_1 X(T_1) + \Psi_1 U \quad (2.12a)$$

$$X(T_3) = \Phi_2 X(T_2) + \Psi_2 U \quad (2.12b)$$

From (2.11) we see that  $X(T_3) = X(T_1)$  in large signal steady state for  $N_s = 2$ , (2.12a) can be substituted into (2.12b) and  $X(T_1)$  can be solved for:

$$X(T_1) = (I - \Phi_2\Phi_1)^{-1}(\Phi_2\Psi_1 + \Psi_2)U \quad (2.13)$$

With  $X(T_1)$  solved for, (2.13) can then be plugged in to (2.12a) to obtain an expression for  $X(T_2)$ . For  $N_s = 3$ , (2.10) becomes:

$$X(T_2) = \Phi_1 X(T_1) + \Psi_1 U \quad (2.14a)$$

$$X(T_3) = \Phi_2 X(T_2) + \Psi_2 U \quad (2.14b)$$

$$X(T_4) = \Phi_3 X(T_3) + \Psi_3 U \quad (2.14c)$$

From (2.11) we see that  $X(T_4) = X(T_1)$  in large signal steady state for  $N_s = 3$ , (2.14a) can be substituted into (2.14b) which can then be substituted into (2.14c).  $X(T_1)$  can then be solved for:

$$X(T_1) = (I - \Phi_3\Phi_2\Phi_1)^{-1}(\Phi_3\Phi_2\Psi_1 + \Phi_3\Psi_2 + \Psi_3)U \quad (2.15)$$

(2.15) can then be substituted into (2.14b) and (2.14c) for expressions for  $X(T_2)$  and  $X(T_3)$  respectively.

To obtain the small signal difference equation, the state vector  $x(t_i)$  and switching time  $t_i$  are first split into their DC and AC components:

$$\begin{aligned} x(t_i) &= X(t_i) + \hat{x}(t_i) \\ &= X(T_i + \hat{t}_i) + \hat{x}(T_i + \hat{t}_i) \end{aligned} \quad (2.16)$$

Taking the Taylor series expansions of the terms in (2.16):

$$\begin{aligned} X(T_i + \hat{t}_i) &\approx X(T_i) + \dot{X}(T_i)\hat{t}_i \\ \hat{x}(T_i + \hat{t}_i) &\approx \hat{x}(T_i) + \dot{\hat{x}}(T_i)\hat{t}_i \end{aligned} \quad (2.17)$$

Combining (2.16) and (2.17):

$$x(t_i) \approx X(T_i) + \dot{X}(T_i)\hat{t}_i + \hat{x}(T_i) + \dot{\hat{x}}(T_i)\hat{t}_i \quad (2.18)$$

Dropping all but the first order (small signal) terms and substituting (2.7a) gives  $\hat{x}(t_i)$  and puts it in a form that will be convenient later on:

$$\begin{aligned} \hat{x}(t_i) &\approx \dot{X}(T_i)\hat{t}_i + \hat{x}(T_i) \\ &= [A_i X(T_i) + B_i U] \hat{t}_i + \hat{x}(T_i) \end{aligned} \quad (2.19)$$

The input is also perturbed to get:

$$u(t) = U + \hat{u}(t) \quad (2.20)$$

where  $U$  is the constant DC component of  $u(t)$  and  $\hat{u}(t)$  is the AC component which will be chosen to take the form of  $\hat{u}(t) = \hat{u}_p e^{j\omega t}$ , where  $\hat{u}_p$  is the peak value of the perturbation. The reasons for choosing an exponential form for the excitation signal is that the real component of  $e^{j\omega t}$  is a cosine wave and the exponential form will be convenient later on in solving the small signal difference equation for  $\hat{x}(T_i)$ . For similar reasons, all perturbations in this thesis will be chosen to be of the same form.  $\omega$  represents the frequency of the perturbation signal in radians/second. Substituting (2.20) into (2.6) gives:

$$\begin{aligned} x(t_{i+1}) &= e^{A_i(t_{i+1}-t_i)}x(t_i) + \int_{t_i}^{t_{i+1}} e^{A_i(t_{i+1}-\tau)} B_i U d\tau \\ &\quad + \int_{t_i}^{t_{i+1}} e^{A_i(t_{i+1}-\tau)} B_i \hat{u}_p e^{j\omega\tau} d\tau \end{aligned} \quad (2.21)$$

Evaluating the integral yields:

$$\begin{aligned} x(t_{i+1}) &= e^{A_i(t_{i+1}-t_i)}x(t_i) + A_i^{-1} [e^{A_i(t_{i+1}-t_i)} - I] B_i U \\ &\quad + (j\omega - A_i)^{-1} e^{j\omega t_{i+1}} [I - e^{-(j\omega I - A_i)(t_{i+1}-t_i)}] B_i \hat{u}_p \end{aligned} \quad (2.22)$$



Substituting  $t_i = T_i + \hat{t}_i$  into (2.22) and denoting  $\chi_i = j\omega I - A_i$  and  $\beta_i = e^{-j\omega(T_{i+1}-T_i)}\Phi_i$  results in:

$$\begin{aligned}
x(t_{i+1}) &= e^{A_i[(T_{i+1}-T_i)+(\hat{t}_{i+1}-\hat{t}_i)]}x(t_i) + A_i^{-1}[e^{A_i[(T_{i+1}-T_i)+(\hat{t}_{i+1}-\hat{t}_i)]} - I]B_i U \\
&\quad + \chi_i^{-1}e^{j\omega(T_{i+1}+\hat{t}_{i+1})}[I - e^{-\chi_i[(T_{i+1}-T_i)+(\hat{t}_{i+1}-\hat{t}_i)]}]B_i \hat{u}_p \\
&= \Phi_i e^{A_i(\hat{t}_{i+1}-\hat{t}_i)}x(t_i) + A_i^{-1}[\Phi_i e^{A_i(\hat{t}_{i+1}-\hat{t}_i)} - I]B_i U \\
&\quad + \chi_i^{-1}e^{j\omega T_{i+1}}e^{j\omega \hat{t}_{i+1}}[I - \beta_i e^{-\chi_i(\hat{t}_{i+1}-\hat{t}_i)}]B_i \hat{u}_p
\end{aligned} \tag{2.23}$$

With time perturbations  $\hat{t}_i$  and  $\hat{t}_{i+1}$  appearing as exponents, the exponentials containing them will be linearized by replacing them with the first two terms of the Taylor series expansion:

$$e^x \approx 1 + x \tag{2.24}$$

which allows (2.23) to take the form:

$$\begin{aligned}
x(t_{i+1}) &= \Phi_i[I + A_i(\hat{t}_{i+1} - \hat{t}_i)]x(t_i) + A_i^{-1}\{\Phi_i[I + A_i(\hat{t}_{i+1} - \hat{t}_i)] - I\}B_i U \\
&\quad + \chi_i^{-1}e^{j\omega T_{i+1}}(1 + j\omega \hat{t}_{i+1})\{I - \beta_i[I + \chi_i(\hat{t}_{i+1} - \hat{t}_i)]\}B_i \hat{u}_p \tag{2.25}
\end{aligned}$$

It should be noted that the application of (2.24) to linearize (2.25) is an approximation however large and small signal models only consider zeroth and first order terms, respectively, thus no accuracy is lost in the "small signal sense". Substituting in (2.18) for  $x(t_i)$  and  $x(t_{i+1})$ , dropping any terms that are not small signal terms (first order), and solving for  $\hat{x}(T_{i+1})$ :

$$\begin{aligned}
\hat{x}(T_{i+1}) &= \Phi_i \hat{x}(T_i) + [(A_i - A_{i+1})X(T_{i+1}) + (B_i - B_{i+1})U]\hat{t}_{i+1} \\
&\quad + \chi_i^{-1}[I - \beta_i]B_i \hat{u}_p e^{j\omega T_{i+1}} \tag{2.26}
\end{aligned}$$

Letting  $\eta_i = \chi_i^{-1}[I - \beta_i]$  and  $\xi_i = (A_i - A_{i+1})X(T_{i+1}) + (B_i - B_{i+1})U$  results in the small signal difference equation:

$$\hat{x}(T_{i+1}) = \Phi_i \hat{x}(T_i) + \xi_i \hat{t}_{i+1} + \eta_i B_i \hat{u}(T_{i+1}) \quad (2.27)$$

The small signal difference equation will later be solved for a finite  $N_s$ .

To summarize, the difference equations developed in this chapter along with some important identities are, for  $i \in (1, 2, \dots, N_s)$ :

DC Large Signal Difference Equation:

$$X(T_{i+1}) = \Phi_i X(T_i) + \Psi_i U \quad (2.10)$$

AC Small Signal Difference Equation:

$$\hat{x}(T_{i+1}) = \Phi_i \hat{x}(T_i) + \xi_i \hat{t}_{i+1} + \eta_i B_i \hat{u}(T_{i+1}) \quad (2.27)$$

where

$$\begin{aligned} \Phi_i &= e^{A_i(T_{i+1}-T_i)} \\ \Psi_i &= A_i^{-1}[\Phi_i - I]B_i \\ \chi_i &= j\omega I - A_i \\ \beta_i &= e^{-j\omega(T_{i+1}-T_i)}\Phi_i \\ \eta_i &= \chi_i^{-1}(I - \beta_i) \\ \xi_i &= (A_i - A_{i+1})X(T_{i+1}) + (B_i - B_{i+1})U \end{aligned} \quad (2.28)$$

Expressions for  $X(T_1)$  for  $N_s = 2$  and  $N_s = 3$ :

$$X(T_1) = (I - \Phi_2\Phi_1)^{-1}(\Phi_2\Psi_1 + \Psi_2)U \quad (2.13)$$

$$X(T_1) = (I - \Phi_3\Phi_2\Phi_1)^{-1}(\Phi_3\Phi_2\Psi_1 + \Phi_3\Psi_2 + \Psi_3)U \quad (2.15)$$

Another useful identity is:

$$\gamma_i = \dot{X}(T_{i+1}) + \xi_i = A_i X(T_{i+1}) + B_i U \quad (2.29)$$

The following scalars or vectors may be split into their DC and AC components:

$$\begin{aligned} t_i &= T_i + \hat{t}_i \\ d &= D + \hat{d}(t) \\ x(t) &= X(t) + \hat{x}(t) \\ u(t) &= U + \hat{u}(t) \\ r(t) &= R + \hat{r}(t) \end{aligned} \quad (2.30)$$

where all perturbation signals take the following form:

$$\begin{aligned} \hat{d}(t) &= \hat{d}_p e^{j\omega t} \\ \hat{u}(t) &= \hat{u}_p e^{j\omega t} \\ \hat{r}(t) &= \hat{r}_p e^{j\omega t} \end{aligned} \quad (2.31)$$

where  $\hat{d}_p$ ,  $\hat{u}_p$ , and  $\hat{r}_p$  are the peak magnitudes of the perturbations. It should be noted that because  $u(t)$  is a scalar containing only  $v_g(t)$ ,  $\hat{u}_p$  is the peak magnitude of  $\hat{v}_g(t)$ . Also, the following holds for a small signal state vector in large signal steady state in the presence of exponential perturbations:

$$\hat{x}(t)e^{j\omega n T_s} = \hat{x}(t + nT_s) \quad (2.32)$$

for all positive  $n$ , which relates  $\hat{x}(t)$  to  $\hat{x}(t + nT_s)$  for all  $t$ .

### Chapter 3: The Exact Small Signal Transfer Function Expression

A general expression for the transfer functions of interest will be derived in this chapter by taking the average of the time varying function of a periodic network. This general expression will represent the frequency response of time interval modulated switched networks. The time varying transfer function is defined as:

$$H(j\omega, t) = \int_{-\infty}^{\infty} h(t, \tau) e^{-j\omega\tau} d\tau \quad (3.1)$$

where  $h(t, \tau)$  is the impulse response at time  $t$  caused by an impulse at time  $t - \tau$ . The output of the system  $y(t)$  can be found by generalizing (3.1) for any input  $u(t)$ :

$$y(t) = \int_{-\infty}^{\infty} h(t, \tau) u(t - \tau) d\tau \quad (3.2)$$

Considering an input of the form  $u(t) = e^{j\omega t}$  as in (2.31) results in:

$$y(t) = \int_{-\infty}^{\infty} h(t, \tau) e^{j\omega(t-\tau)} d\tau = \left[ \int_{-\infty}^{\infty} h(t, \tau) e^{-j\omega\tau} d\tau \right] e^{j\omega t} = H(j\omega, t) e^{j\omega t} \quad (3.3)$$

Rearranging gives a useful definition that will be used later:

$$H(j\omega, t) = \left. \frac{y(t)}{u(t)} \right|_{u(t)=e^{j\omega t}} \quad (3.4)$$

The relation between  $u(t)$  and its Fourier transform  $U(j\omega)$  is:

$$u(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} U(j\omega) e^{j\omega t} d\omega \quad (3.5)$$

where

$$U(j\omega) = 2\pi\delta(\omega) \quad (3.6)$$

and  $\delta(\omega)$  is the Dirac delta function. Substituting (3.5) into (3.2) yields:

$$\begin{aligned}
y(t) &= \int_{-\infty}^{\infty} h(t, \tau) \left[ \frac{1}{2\pi} \int_{-\infty}^{\infty} U(j\omega) e^{j\omega(t-\tau)} d\omega \right] d\tau \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} h(t, \tau) e^{-j\omega\tau} d\tau \right] U(j\omega) e^{j\omega t} d\omega \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} H(j\omega, t) U(j\omega) e^{j\omega t} d\omega
\end{aligned} \tag{3.7}$$

From (3.7) the output frequency spectrum  $Y(j\omega)$  can be found. In taking the Fourier transform of (3.7), a dummy variable  $\mu$  will be used to avoid confusion:

$$\begin{aligned}
Y(j\mu) &= \int_{-\infty}^{\infty} y(t) e^{-j\mu t} dt \\
&= \int_{-\infty}^{\infty} \left[ \frac{1}{2\pi} \int_{-\infty}^{\infty} H(j\omega, t) U(j\omega) e^{j\omega t} d\omega \right] e^{-j\mu t} dt \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} H(j\omega, t) e^{j\omega t} e^{-j\mu t} dt \right] U(j\omega) d\omega \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(j\omega, t) e^{j\omega t} \delta(\omega) d\omega e^{-j\mu t} dt
\end{aligned} \tag{3.8}$$

After applying the sifting property to (3.8):

$$Y(j\mu) = \int_{-\infty}^{\infty} H(0, t) e^{-j\mu t} dt \tag{3.9}$$

Partitioning the time axis into an infinite number of switching periods indexed by  $k$  and then further subdividing the switching periods into  $N_s$  subintervals of length  $t_{i+1} - t_i$  results in:

$$Y(j\mu) = \sum_{k=-\infty}^{\infty} \sum_{i=1}^{N_s} \int_{t_i}^{t_{i+1}} H(0, t) e^{-j\mu t} dt \tag{3.10}$$

Noting from (2.4ab) and (3.4) that:

$$H(0, t) = y(t) = C_i^T x(t) + E_i u(t) \quad (3.11)$$

(3.10) becomes:

$$Y(j\mu) = \sum_{k=-\infty}^{\infty} \sum_{i=1}^{N_s} \int_{t_i}^{t_{i+1}} [C_i^T x(t) + E_i u(t)] e^{-j\mu t} dt \quad (3.12)$$

Substituting in (2.5) and (2.20) into (3.12), evaluating the integral, linearizing the terms with  $\hat{t}_i$  and  $\hat{t}_{i+1}$ , and dropping all of the terms that are not small signal terms (first order) results in:

$$\hat{Y}(j\omega) = \sum_{k=-\infty}^{\infty} \sum_{i=1}^{N_s} e^{-j\omega T_i} [C_i^T \eta_i \hat{x}(T_i) + \kappa_i \hat{t}_i + \lambda_i \hat{t}_{i+1} + \rho_i \hat{u}(T_i)] \quad (3.13)$$

where

$$\begin{aligned} \eta_i &= \chi_i^{-1} (I - \beta_i) \\ \kappa_i &= -[C_i^T X(T_i) + E_i U] \\ \lambda_i &= [C_i^T X(T_{i+1}) + E_i U] e^{j\omega(T_{i+1} - T_i)} \\ \rho_i &= C_i^T \chi_i^{-1} [(T_{i+1} - T_i)I - \eta_i] B_i + E_i (T_{i+1} - T_i) \end{aligned} \quad (3.14)$$

The expression (3.13) can be simplified by examining the terms of the summation regarding  $\hat{t}_i$  and  $\hat{t}_{i+1}$ :

$$\sum_{i=1}^{N_s} e^{-j\omega T_i} (\kappa_i \hat{t}_i + \lambda_i \hat{t}_{i+1}) \quad (3.15)$$

The summation (3.15) can be expanded into:

$$\sum_{i=1}^{N_s} e^{-j\omega T_i} (\kappa_i \hat{t}_i + \lambda_i \hat{t}_{i+1}) = e^{-j\omega T_i} \kappa_i \hat{t}_i + e^{-j\omega T_{N_s}} \lambda_{N_s} \hat{t}_{N_s} + \sum_{i=1}^{N_s-1} (e^{-j\omega T_i} \lambda_i + e^{-j\omega T_{i+1}} \kappa_{i+1}) \hat{t}_{i+1} \quad (3.16)$$

Further expanding the first two terms of (3.16):

$$\begin{aligned} e^{-j\omega T_i} \kappa_i \hat{t}_i + e^{-j\omega T_{N_s}} \lambda_{N_s} \hat{t}_{N_s} &= e^{-j\omega T_i} \{ -[C_1^T X(T_i) + E_1 U] \} \hat{t}_1 \\ &+ e^{-j\omega T_{N_s}} [C_{N_s}^T X(T_{N_s+1}) + E_{N_s} U] e^{-j\omega (T_{N_s+1} - T_{N_s})} \hat{t}_{N_s+1} \end{aligned} \quad (3.17)$$

Because the system is periodic and cycles through  $N_s$  topologies, the following relations hold:

$$\begin{aligned} C_1^T &= C_{N_s+1}^T \\ E_1 &= E_{N_s+1} \\ \hat{t}_1 &= \hat{t}_{N_s+1} e^{-j\omega (T_{N_s+1} - T_1)} \\ X(T_1) &= X(T_{N_s+1}) \end{aligned} \quad (3.18)$$

Letting  $\zeta_i = (C_i^T - C_{i+1}^T)X(T_{i+1}) + (E_i - E_{i+1})U$ , (3.17) becomes:

$$e^{-j\omega T_i} \kappa_i \hat{t}_i + e^{-j\omega T_{N_s}} \lambda_{N_s} \hat{t}_{N_s} = e^{-j\omega T_{N_s+1}} \zeta_{N_s} \hat{t}_{N_s+1} \quad (3.19)$$

and the last term of (3.16) becomes:

$$\sum_{i=1}^{N_s-1} (e^{-j\omega T_i} \lambda_i + e^{-j\omega T_{i+1}} \kappa_{i+1}) \hat{t}_{i+1} = \sum_{i=1}^{N_s-1} e^{-j\omega T_{i+1}} \zeta_i \hat{t}_{i+1} \quad (3.20)$$

Combining (3.19) and (3.20) allows (3.15) to be expressed as:

$$\sum_{i=1}^{N_s} e^{-j\omega T_i} (\kappa_i \hat{t}_i + \lambda_i \hat{t}_{i+1}) = \sum_{i=1}^{N_s} e^{-j\omega T_{i+1}} \zeta_i \hat{t}_{i+1} \quad (3.21)$$

and (3.13) to take the final simplified form:

$$\hat{Y}(j\omega) = \sum_{k=-\infty}^{\infty} \sum_{i=1}^{N_s} e^{-j\omega T_i} [C_i^T \eta_i \hat{x}(T_i) + e^{-j\omega(T_{i+1}-T_i)} \zeta_i \hat{t}_{i+1} + \rho_i \hat{u}(T_i)] \quad (3.22)$$

While  $\hat{Y}(j\omega)$  is the entire output spectrum containing impulses at integer multiples of  $\omega$ , we are only interested in the results at  $\omega$  which is the frequency response  $H_0(j\omega)$ . Reference [1] isolates  $H_0(j\omega)$  by taking the average of the inner summation over a period  $T_x$  rather than the entire output spectrum. Fig. 7 shows that  $T_x$  can be an integer  $N$  multiple of  $T_s$ , therefore:

$$H_0(j\omega) = \frac{1}{NT_s} \sum_{k=0}^{N-1} \sum_{i=1}^{N_s} e^{-j\omega T_i} [C_i^T \eta_i \hat{x}(T_i) + e^{-j\omega(T_{i+1}-T_i)} \zeta_i \hat{t}_{i+1} + \rho_i \hat{u}(T_i)] \quad (3.23)$$

While it appears from (3.23) that the outer summation will have to be evaluated  $N$  times since there are small signal terms that differ between switching cycles, after substitution and evaluation of the inner summation it is evident, for all of the cases investigated in this thesis, that these terms will become independent of which  $T_s$  they are evaluated for. Therefore, the outer summation will result in  $N$  equivalent terms which are then divided by  $N$ . This is equivalent to setting  $N = 1$ , which is valid for the cases in this thesis.

It will become evident that finding the transfer functions of interest involves first determining expressions for  $\hat{t}_i$  for  $i \in (1, 2, \dots, N_s)$ , solving the difference equation to obtain an expression for  $\hat{x}(T_1)$ , and then substituting the results into the general expression  $H_0(j\omega)$ . After extracting  $\frac{\hat{v}_{out}}{\hat{r}}$  and  $\frac{\hat{v}_{out}}{\hat{v}_g}$  from  $H_0(j\omega)$  the Bode plots of the transfer functions will be compared to those from simulation.



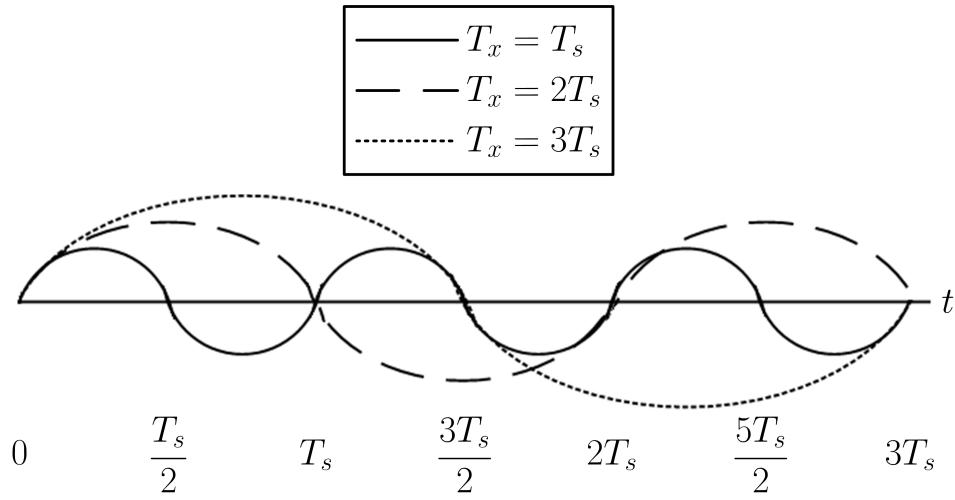


Figure 7: Period  $T_x$  in the presence of perturbations, where the waveform of period  $T_s$  can represent any output of the system while in steady state. While the shortest possible  $T_x$  is not necessarily an integer  $N$  multiple of  $T_s$ , the relation  $T_x = NT_s$  does provide convenience.

## Chapter 4: Normal Voltage Mode Transfer Functions

The transfer functions of interest will first be derived for the NVM boost converter shown in Fig. 8 for operation in CCM followed by DCM. The NVM OTB converter shown in Fig. 9 will have its transfer functions derived for operation in DVM, an operating condition not applicable to the boost converter. The output voltage controller has been omitted in Fig. 8 and Fig. 9 since the open loop transfer functions are derived, so the control signal  $r(t)$  is externally supplied. For both converters  $r(t)$  is compared to a sawtooth signal in order to generate a square wave to directly control  $Q$  and regulate the switching between topologies. A NVM converter has its sawtooth signal externally supplied rather than being generated from some output of the system. Although it is preferable to have the sawtooth signal generated from the system, since doing so reduces the effect of perturbations on output voltage  $v_{out}(t)$ , NVM is the simplest implementation and should therefore be considered.

So far, the start of the switching period  $T_s$  has been defined by the period of the sawtooth signal. This way of defining the beginning of the  $T_s$  will be called Model 1. Defining the start of the interval to be the switching event controlled by  $r(t)$  instead, which will be called Model 2, will be considerably more convenient. For this reason, the derivations will take place in Model 2, and then the subscripts of the results rotated appropriately so that they apply to Model 1. Fig. 10 shows very generally how Model 1 and Model 2 are related. While [2] and [3] have previously derived the exact transfer functions for a NVM converter in CCM and  $\frac{\hat{v}_{out}}{\hat{r}}$  for a NVM converter in DCM,  $\frac{\hat{v}_{out}}{\hat{u}}$  for a NVM converter in DCM as well as the exact transfer functions of a NVM converter in DVM have yet to be published.

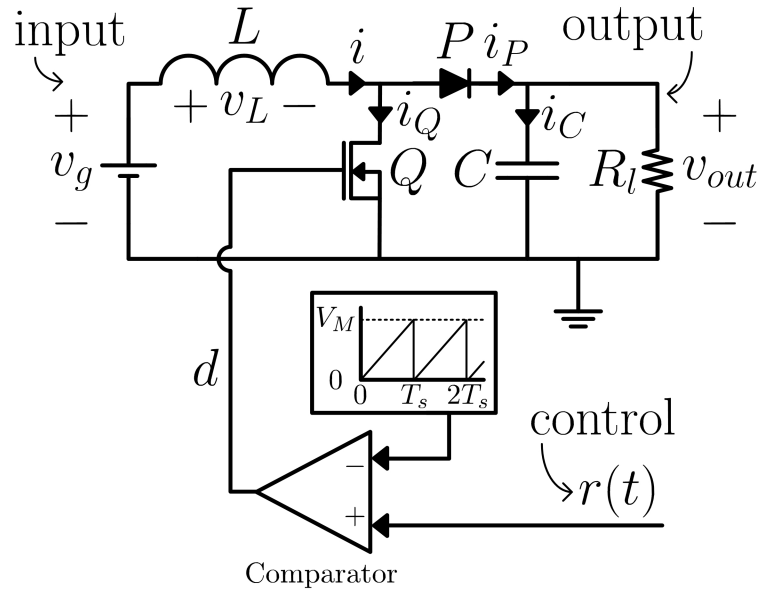


Figure 8: A NVM boost converter. The control signal  $r(t)$ , externally supplied in this figure, is compared to an externally supplied sawtooth signal of amplitude  $V_M$  and switching period  $T_s$  to create a square wave with a duty ratio  $d$  that regulates the switching between topologies.

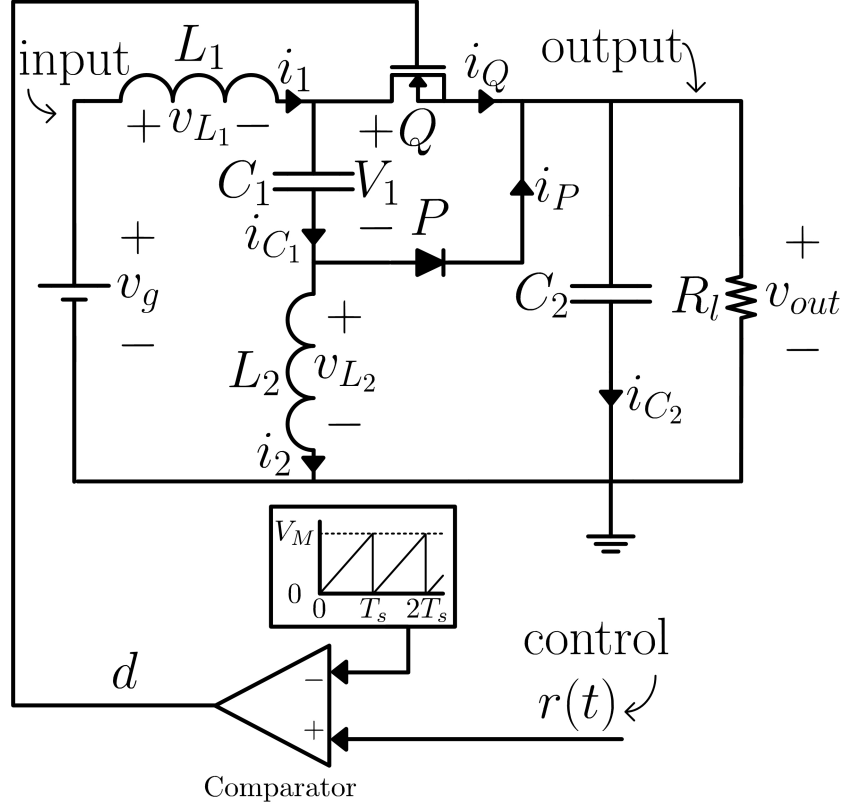


Figure 9: A NVM OTB converter. The control signal  $r(t)$ , externally supplied in this figure, is compared to an externally supplied sawtooth signal of amplitude  $V_M$  and switching period  $T_s$  to create a square wave with a duty ratio  $d$  that regulates the switching between topologies.

#### Section 4.1: Normal Voltage Mode Converter in CCM

A power converter operating in CCM implies that there are two topologies ( $N_s = 2$ ) and thus two subintervals within a switching period  $T_s$ . Considering a CCM converter with Model 1, the first subinterval has an average (DC) length  $DT_s$  and the second interval has an DC length  $(1 - D)T_s$ . The effective topologies of the boost power stage during the two subintervals are shown in Fig. 11. The switching period is described by Fig. 12 in both Model 1 and Model 2. Model 2 will be used for the following derivation, effectively swapping the order of the two subintervals.

Considering Model 2, the switching time  $t_2$  is determined by the period  $T_s$  of the sawtooth signal and thus  $\hat{t}_2 = 0$ .  $t_3$  is the time when the sawtooth signal reaches the value of  $r(t)$ :

$$t_3 = \frac{T_s r(t_3)}{V_M} \quad (4.1)$$

Dropping the DC components of  $r(t_3)$  and  $t_3$ :

$$\hat{t}_3 = \frac{T_s \hat{r}(t_3)}{V_M} \quad (4.2)$$

Taking the Taylor series expansion of  $\hat{r}(t_3)$  and dropping the second order term:

$$\begin{aligned} \hat{r}(t_3) &= \hat{r}(T_3 + \hat{t}_3) \\ &\approx \hat{r}(T_3) + \dot{\hat{r}}(T_3)\hat{t}_3 \\ &\approx \hat{r}(T_3) \end{aligned} \quad (4.3)$$

Substituting (4.3) into (4.2) gives an expression for  $\hat{t}_3$ :

$$\hat{t}_3 = \frac{T_s \hat{r}(T_3)}{V_M} = \mu \hat{r}(T_3) \quad (4.4)$$

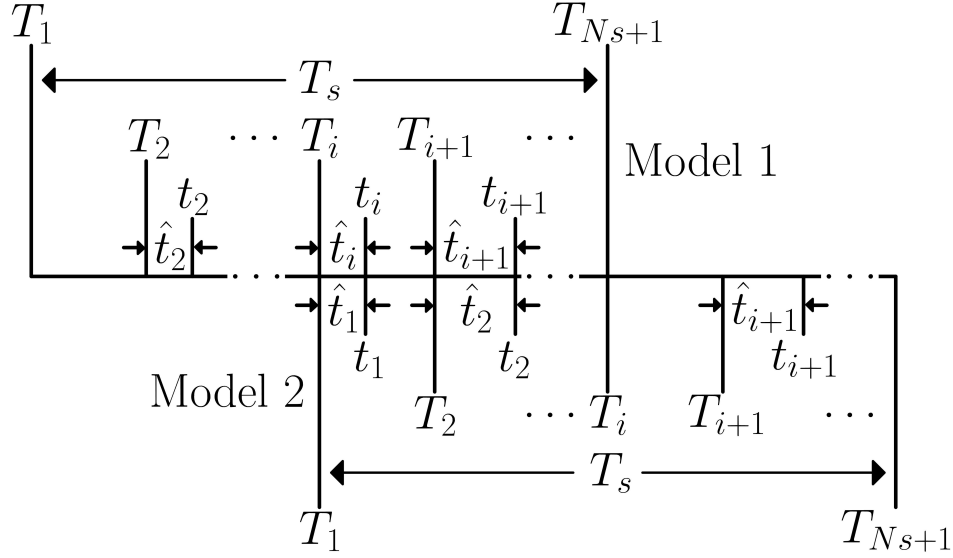


Figure 10: A general switching period  $T_s$  described under Model 1 and Model 2. Under Model 1 the start and end of  $T_s$  is defined by the switching period of the sawtooth signal, thus  $\hat{t}_1 = \hat{t}_{N_s+1} = 0$ , and the switching event  $t_i$  is controlled by  $r(t)$ . Under Model 2, the start and end of  $T_s$  is the switching event determined by  $r(t)$ .

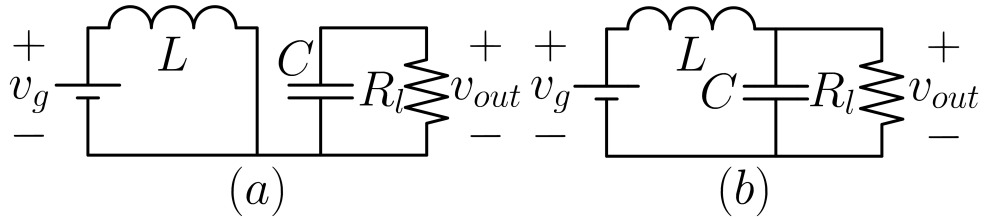


Figure 11: The effective topologies of a NVM boost converter in CCM assuming ideal switches. (a) is the effective topology during the first subinterval of DC length  $DT_s$  where  $Q$  is on and  $P$  is off. (b) is the effective topology during the second subinterval of DC length  $(1 - D)T_s$  where  $Q$  is off and  $P$  is on.

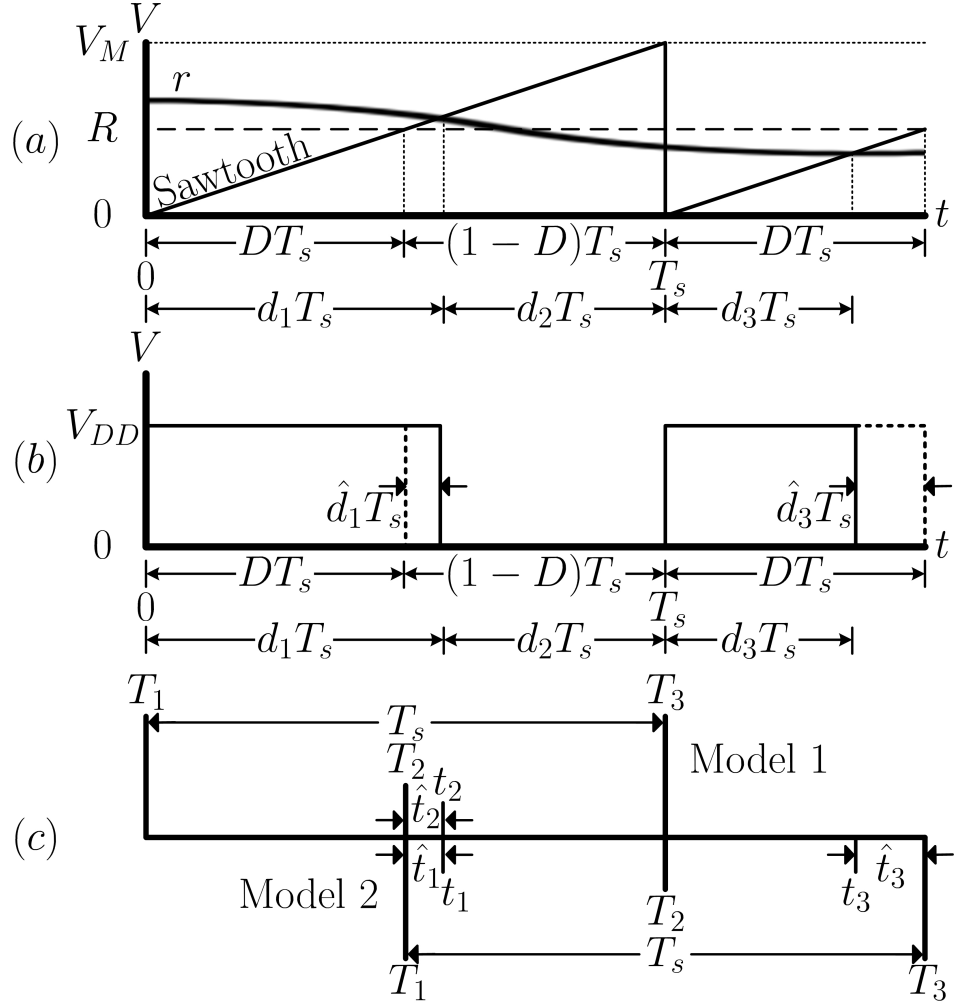


Figure 12: Switching period  $T_s$  for a NVM converter in CCM, described for both Model 1 and Model 2. (a) shows the sawtooth voltage of period  $T_s$  being compared to the control signal  $r(t)$  to produce the square wave (b). (c) illustrates how the times are denoted under Model 1 and Model 2.

where

$$\mu = \frac{T_s}{V_M} \quad (4.5)$$

With  $\hat{t}_3$  obtained, the next step is to get  $\hat{x}(T_1)$  by solving the difference equation (2.27). From (2.27) an expression for  $\hat{x}(T_2)$  is found:

$$\hat{x}(T_2) = \Phi_1 \hat{x}(T_1) + \eta_1 B_1 \hat{u}(T_2) \quad (4.6)$$

Using (2.27) again and substituting in (4.4) and (4.6) gives:

$$\begin{aligned} \hat{x}(T_3) &= \Phi_2 \hat{x}(T_2) + \xi_2 \hat{t}_3 + \eta_2 B_2 \hat{u}(T_3) \\ &= \Phi_2 \Phi_1 \hat{x}(T_1) + \xi_2 \mu \hat{r}(T_3) + \Phi_2 \eta_1 B_1 \hat{u}(T_2) + \eta_2 B_2 \hat{u}(T_3) \end{aligned} \quad (4.7)$$

Expressing all of the signals in their exponential form as in (2.31) and substituting in (2.32) for  $\hat{x}(T_3)$ :

$$\hat{x}(T_1) e^{j\omega T_s} = \Phi_2 \Phi_1 \hat{x}(T_1) + \xi_2 \mu \hat{r}_p e^{j\omega T_3} + \Phi_2 \eta_1 B_1 \hat{u}_p e^{j\omega T_2} + \eta_2 B_2 \hat{u}_p e^{j\omega T_3} \quad (4.8)$$

Solving for  $\hat{x}(T_1)$ :

$$\hat{x}(T_1) = (I - \beta_2 \beta_1)^{-1} [\xi_2 \mu \hat{r}(T_1) + (\beta_2 \eta_1 B_1 + \eta_2 B_2) \hat{u}(T_1)] \quad (4.9)$$

Now all that is left is to substitute (4.4), (4.6), and (4.9) into (3.23).

Expanding (3.23) for  $N_s = 2$  gives:

$$H_0(j\omega) = \frac{1}{T_s} [C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \rho_1 \hat{u}_p + C_2^T \eta_2 \hat{x}(T_2) e^{-j\omega T_2} + \zeta_2 \hat{t}_3 e^{-j\omega T_3} + \rho_2 \hat{u}_p] \quad (4.10)$$

Substituting in (4.4):



$$H_0(j\omega) = \frac{1}{T_s} [C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \zeta_2 \mu \hat{r}_p + C_2^T \eta_2 \hat{x}(T_2) e^{-j\omega T_2} + (\rho_1 + \rho_2) \hat{u}_p] \quad (4.11)$$

Now substituting in (4.6):

$$H_0(j\omega) = \frac{1}{T_s} [(C_1^T \eta_1 + C_2^T \eta_2 \beta_1) \hat{x}(T_1) e^{-j\omega T_1} + \zeta_2 \mu \hat{r}_p + (C_2^T \eta_2 \eta_1 B_1 + \rho_1 + \rho_2) \hat{u}_p] \quad (4.12)$$

Finally substituting in (4.9):

$$\begin{aligned} H_0(j\omega) = \frac{1}{T_s} \{ & [(C_1^T \eta_1 + C_2^T \eta_2 \beta_1)(I - \beta_2 \beta_1)^{-1} \xi_2 \mu + \zeta_2 \mu] \hat{r}_p \\ & + [(C_1^T \eta_1 + C_2^T \eta_2 \beta_1)(I - \beta_2 \beta_1)^{-1} (\beta_2 \eta_1 B_1 + \eta_2 B_2) \\ & + C_2^T \eta_2 \eta_1 B_1 + \rho_1 + \rho_2] \hat{u}_p \} \end{aligned} \quad (4.13)$$

Rotating the subscripts of (4.13) to Model 1 only involves switching them, giving the final result:

$$\begin{aligned} H_0(j\omega) = \frac{1}{T_s} \{ & [(C_1^T \eta_1 \beta_2 + C_2^T \eta_2)(I - \beta_1 \beta_2)^{-1} \xi_1 \mu + \zeta_1 \mu] \hat{r}_p \\ & + [(C_1^T \eta_1 \beta_2 + C_2^T \eta_2)(I - \beta_1 \beta_2)^{-1} (\eta_1 B_1 + \beta_1 \eta_2 B_2) \\ & + C_1^T \eta_1 \eta_2 B_2 + \rho_1 + \rho_2] \hat{u}_p \} \end{aligned} \quad (4.14)$$

where

$$\mu = \frac{T_s}{V_M} \quad (4.15)$$

From (4.14) the transfer functions are extracted by setting the input of interest to one while setting the other to zero. The control-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{r}}$  is therefore found by setting  $\hat{r}_p = 1$  and setting  $\hat{u}_p = 0$ :

$$\frac{\hat{v}_{out}}{\hat{r}} = \frac{1}{V_M} [(C_1^T \eta_1 \beta_2 + C_2^T \eta_2)(I - \beta_1 \beta_2)^{-1} \xi_1 + \zeta_1] \quad (4.16)$$

The input-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{v}_g}$  is found by setting  $\hat{r}_p = 0$  and setting  $\hat{u}_p = 1$ :

$$\begin{aligned} \frac{\hat{v}_{out}}{\hat{v}_g} = \frac{1}{T_s} & [(C_1^T \eta_1 \beta_2 + C_2^T \eta_2)(I - \beta_1 \beta_2)^{-1}(\eta_1 B_1 + \beta_1 \eta_2 B_2) \\ & + C_1^T \eta_1 \eta_2 B_2 + \rho_1 + \rho_2] \quad (4.17) \end{aligned}$$

The control-to-output and input-to-output transfer functions are compared with results from simulation in Fig. 13 and Fig. 14, respectively, with the following parameters:

$$\begin{aligned} T_s &= 10\mu s \\ V_g &= 15V \\ D &= 0.25 \\ R_l &= 18.6\Omega \end{aligned} \quad (4.18)$$

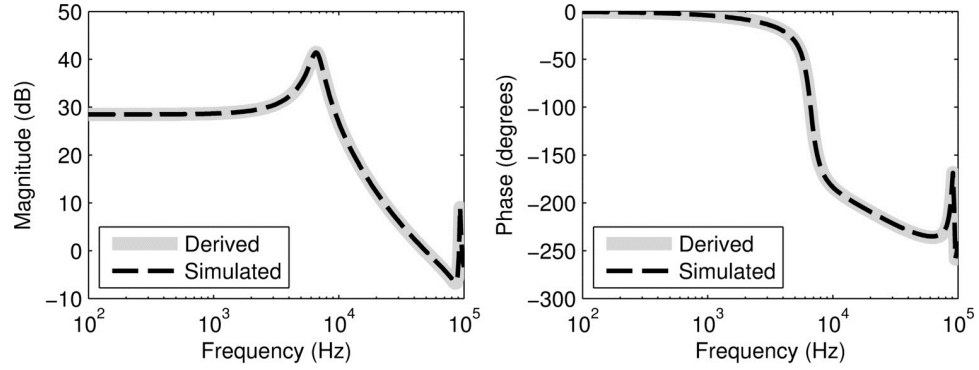


Figure 13: The derived NVM boost converter in CCM control-to-output transfer function (solid) compared with results obtained from simulation (dashed).

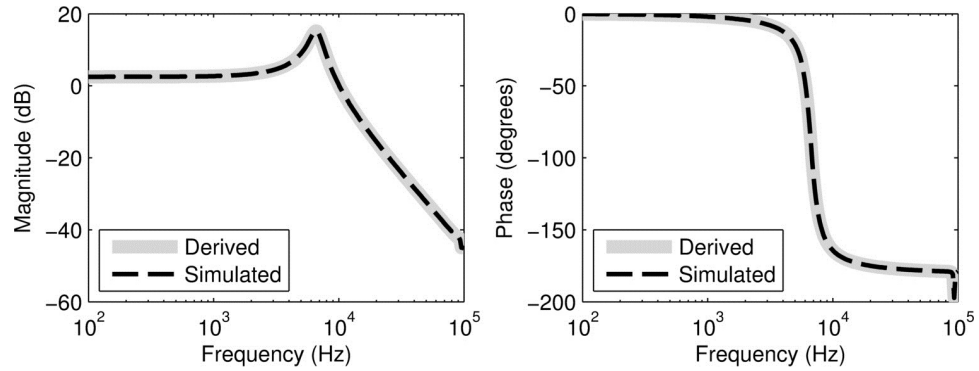


Figure 14: The derived NVM boost converter in CCM input-to-output transfer function (solid) compared with results obtained from simulation (dashed).

## Section 4.2: Normal Voltage Mode Converter in DCM

A power converter operates in DCM when the current through the switching diode  $i_P(t)$  goes to zero, creating a third topology where both switches are turned off. While DCM is generally avoided as it makes  $v_{out}(t)$  dependent on the load resistance  $R_l$ , a large  $R_l$  can force a converter into DCM. DCM can also have desirable damping effects. For  $N_s = 3$ , the switching period  $T_s$  is split into three subintervals of DC lengths  $D_1T_s$ ,  $D_2T_s$ , and  $D_3T_s$ . The effective topologies of the boost power stage during the three subintervals are shown in Fig. 15. The switching period is described by Fig. 16 for both Model 1 and Model 2, the latter of which will be used for the following derivations.

Considering Model 2, the switching time  $t_2$  is the time when:

$$i_P(t_2) = 0 \quad (4.19)$$

Defining a vector  $h^T$  such that:

$$h^T x(t) = i_P(t) \quad (4.20)$$

allows (4.19) to be expressed as:

$$h^T x(t_2) = 0 \quad (4.21)$$

Dropping the DC component of  $x(t_2)$ :

$$h^T \hat{x}(t_2) = 0 \quad (4.22)$$

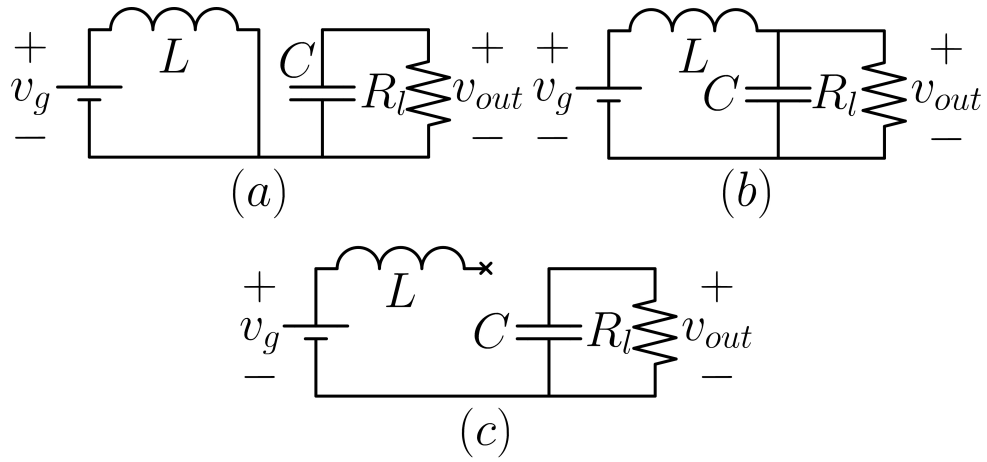


Figure 15: The effective topologies of a NVM boost converter in DCM assuming ideal switches. (a) is the effective topology during the first subinterval of DC length  $D_1T_s$  where  $Q$  is on and  $P$  is off. (b) is the effective topology during the second subinterval of DC length  $D_2T_s$  where  $Q$  is off and  $P$  is on. (c) is the effective topology during the third subinterval of DC length  $D_3T_s$  where both  $Q$  and  $P$  are off.

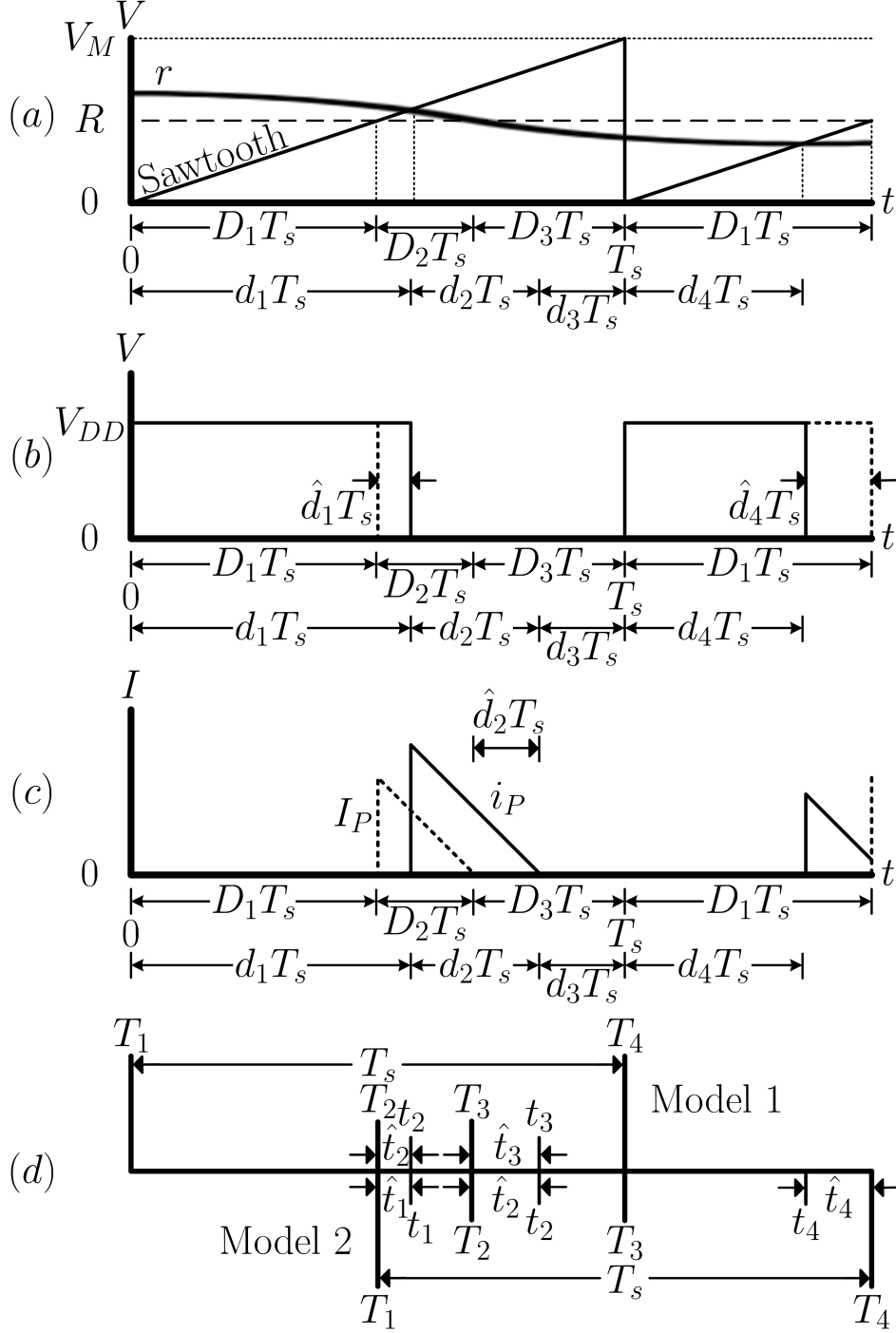


Figure 16: Switching period  $T_s$  for a NVM converter in DCM, described for both Model 1 and Model 2. (a) shows the sawtooth voltage of period  $T_s$  being compared to the control signal  $r(t)$  to produce the square wave of (b). (c) shows the diode current  $i_P(t)$  reaching zero and creating a third topology. (d) illustrates how the times are denoted under Model 1 and Model 2.

Substituting in (2.19) and solving for  $\hat{t}_2$ :

$$\hat{t}_2 = -\mu_1 h^T \Phi_1 \hat{x}(T_1) - \mu_1 h^T \eta_1 B_1 \hat{u}(T_2) \quad (4.23)$$

where

$$\mu_1 = \frac{1}{h^T \gamma_1} \quad (4.24)$$

The switching time  $t_3$  is determined by the switching period of the sawtooth signal and is not sensitive to variations in  $r(t)$  or  $u(t)$ , therefore  $\hat{t}_3 = 0$ . Finding an expression for  $\hat{t}_4$  follows the same procedure as finding  $\hat{t}_3$  for a NVM converter in CCM:

$$\hat{t}_4 = \frac{T_s \hat{r}(T_4)}{V_M} = \mu_3 \hat{r}(T_4) \quad (4.25)$$

where

$$\mu_3 = \frac{T_s}{V_M} \quad (4.26)$$

After getting an expression for  $\hat{t}_4$ ,  $\hat{x}(T_1)$  will be found next. Using (2.27) and substituting in (4.23), an expression for  $\hat{x}(T_2)$  is found:

$$\begin{aligned} \hat{x}(T_2) &= \Phi_1 \hat{x}(T_1) + \xi_1 \hat{t}_2 + \eta_1 B_1 \hat{u}(T_2) \\ &= \Gamma_1 \Phi_1 \hat{x}(T_1) + \Gamma_1 \eta_1 B_1 \hat{u}(T_2) \end{aligned} \quad (4.27)$$

where

$$\Gamma_1 = I - \xi_1 \mu_1 h^T \quad (4.28)$$

Using (2.27) again and substituting in (4.27) gets the following expression for  $\hat{x}(T_3)$ :

$$\begin{aligned}
\hat{x}(T_3) &= \Phi_2 \hat{x}(T_2) + \eta_2 B_2 \hat{u}(T_3) \\
&= \Phi_2 \Gamma_1 \Phi_1 \hat{x}(T_1) + \Phi_2 \Gamma_1 \eta_1 B_1 \hat{u}(T_2) + \eta_2 B_2 \hat{u}(T_3)
\end{aligned} \tag{4.29}$$

Using (2.27) once more and substituting in (4.29) gives an expression for  $\hat{x}(T_4)$ :

$$\begin{aligned}
\hat{x}(T_4) &= \Phi_3 \hat{x}(T_3) + \xi_3 \hat{t}_4 + \eta_3 B_3 \hat{u}(T_4) \\
&= \Phi_3 \Phi_2 \Gamma_1 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}(T_4) + \Phi_3 \Phi_2 \Gamma_1 \eta_1 B_1 \hat{u}(T_2) \\
&\quad + \Phi_3 \eta_2 B_2 \hat{u}(T_3) + \eta_3 B_3 \hat{u}(T_4)
\end{aligned} \tag{4.30}$$

Expressing all of the signals in their exponential form as in (2.31) and substituting in (2.32) for  $\hat{x}(T_4)$ :

$$\begin{aligned}
\hat{x}(T_1) e^{j\omega T_s} &= \Phi_3 \Phi_2 \Gamma_1 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}_p e^{j\omega T_4} + \Phi_3 \Phi_2 \Gamma_1 \eta_1 B_1 \hat{u}_p e^{j\omega T_2} \\
&\quad + \Phi_3 \eta_2 B_2 \hat{u}_p e^{j\omega T_3} + \eta_3 B_3 \hat{u}_p e^{j\omega T_4}
\end{aligned} \tag{4.31}$$

Solving for  $\hat{x}(T_1)$ :

$$\begin{aligned}
\hat{x}(T_1) &= (I - \beta_3 \beta_2 \Gamma_1 \beta_1)^{-1} \{ \xi_3 \mu_3 \hat{r}(T_1) + [\beta_3 (\beta_2 \Gamma_1 \eta_1 B_1 + \eta_2 B_2) \\
&\quad + \eta_3 B_3] \hat{u}(T_1) \}
\end{aligned} \tag{4.32}$$

Substituting (4.25), (4.27), (4.29), and (4.32) into (3.23) is the final step. Expanding (3.23) for  $N_s = 3$  gives:

$$\begin{aligned}
H_0(j\omega) &= \frac{1}{T_s} [C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \zeta_1 \hat{t}_2 e^{-j\omega T_2} + \rho_1 \hat{u}_p \\
&\quad + C_2^T \eta_2 \hat{x}(T_2) e^{-j\omega T_2} + \rho_2 \hat{u}_p + C_3^T \eta_3 \hat{x}(T_3) e^{-j\omega T_3} \\
&\quad + \zeta_3 \hat{t}_4 e^{-j\omega T_4} + \rho_3 \hat{u}_p]
\end{aligned} \tag{4.33}$$

Substituting in (4.23), (4.25), (4.27), and (4.29):



$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left( \{C_1^T \eta_1 + [(C_2^T \eta_2 + C_3^T \eta_3 \beta_2) \Gamma_1 - \zeta_1 \mu_1 h^T] \beta_1\} \right. \\
& \times \hat{x}(T_1) e^{-j\omega T_1} + \zeta_3 \mu_3 \hat{r}_p + \{[(C_2^T \eta_2 + C_3^T \eta_3 \beta_2) \Gamma_1 \\
& \left. - \zeta_1 \mu_1 h^T] \eta_1 B_1 + C_3^T \eta_3 \eta_2 B_2 + \rho_1 + \rho_2 + \rho_3\} \hat{u}_p \right) \quad (4.34)
\end{aligned}$$

Finally substituting in (4.32):

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left[ \left( \{C_1^T \eta_1 + [(C_2^T \eta_2 + C_3^T \eta_3 \beta_2) \Gamma_1 - \zeta_1 \mu_1 h^T] \beta_1\} \right. \right. \\
& \times (I - \beta_3 \beta_2 \Gamma_1 \beta_1)^{-1} \xi_3 \mu_3 + \zeta_3 \mu_3 \Big) \hat{r}_p + \left( \{C_1^T \eta_1 + [(C_2^T \eta_2 \right. \\
& + C_3^T \eta_3 \beta_2) \Gamma_1 - \zeta_1 \mu_1 h^T] \beta_1\} (I - \beta_3 \beta_2 \Gamma_1 \beta_1)^{-1} [\beta_3 (\beta_2 \Gamma_1 \eta_1 B_1 \\
& + \eta_2 B_2) + \eta_3 B_3] + [(C_2^T \eta_2 + C_3^T \eta_3 \beta_2) \Gamma_1 - \zeta_1 \mu_1 h^T] \eta_1 B_1 \\
& \left. \left. + C_3^T \eta_3 \eta_2 B_2 + \rho_1 + \rho_2 + \rho_3 \right) \hat{u}_p \right] \quad (4.35)
\end{aligned}$$

Rotating the subscripts of (4.35) to Model 1 involves replacing 1 with 2, 2 with 3, and 3 with 1. The final result is:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left[ \left( \{C_2^T \eta_2 + [(C_3^T \eta_3 + C_1^T \eta_1 \beta_3) \Gamma_2 - \zeta_2 \mu_2 h^T] \beta_2\} \right. \right. \\
& \times (I - \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} \xi_1 \mu_1 + \zeta_1 \mu_1 \Big) \hat{r}_p + \left( \{C_2^T \eta_2 + [(C_3^T \eta_3 \right. \\
& + C_1^T \eta_1 \beta_3) \Gamma_2 - \zeta_2 \mu_2 h^T] \beta_2\} (I - \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} [\beta_1 (\beta_3 \Gamma_2 \eta_2 B_2 \\
& + \eta_3 B_3) + \eta_1 B_1] + [(C_3^T \eta_3 + C_1^T \eta_1 \beta_3) \Gamma_2 - \zeta_2 \mu_2 h^T] \eta_2 B_2 \\
& \left. \left. + C_1^T \eta_1 \eta_3 B_3 + \rho_1 + \rho_2 + \rho_3 \right) \hat{u}_p \right] \quad (4.36)
\end{aligned}$$

where

$$\begin{aligned}
\mu_1 &= \frac{T_s}{V_M} \\
\mu_2 &= \frac{1}{h^T \gamma_2}
\end{aligned} \quad (4.37)$$

Extracting the transfer functions from (4.36) is accomplished by setting the appropriate input to one and the others to zero. The control-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{r}}$  is then found by setting  $\hat{r}_p = 1$  and setting  $\hat{u}_p = 0$ :

$$\begin{aligned} \frac{\hat{v}_{out}}{\hat{r}} = \frac{1}{V_M} & \left( \{C_2^T \eta_2 + [(C_3^T \eta_3 + C_1^T \eta_1 \beta_3) \Gamma_2 - \zeta_2 \mu_2 h^T] \beta_2\} \right. \\ & \left. \times (I - \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} \xi_1 + \zeta_1 \right) \quad (4.38) \end{aligned}$$

The input-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{v}_g}$  is found by setting  $\hat{r}_p = 0$  and  $\hat{u}_p = 1$ :

$$\begin{aligned} \frac{\hat{v}_{out}}{\hat{v}_g} = \frac{1}{T_s} & \left( \{C_2^T \eta_2 + [(C_3^T \eta_3 + C_1^T \eta_1 \beta_3) \Gamma_2 - \zeta_2 \mu_2 h^T] \beta_2\} \right. \\ & \times (I - \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} [\beta_1 (\beta_3 \Gamma_2 \eta_2 B_2 + \eta_3 B_3) + \eta_1 B_1] + [(C_3^T \eta_3 \\ & \left. + C_1^T \eta_1 \beta_3) \Gamma_2 - \zeta_2 \mu_2 h^T] \eta_2 B_2 + C_1^T \eta_1 \eta_3 B_3 + \rho_1 + \rho_2 + \rho_3 \right) \quad (4.39) \end{aligned}$$

The control-to-output and input-to-output transfer functions are compared with results from simulation in Fig. 17 and Fig. 18, respectively, with the following parameters:

$$\begin{aligned} T_s &= 10 \mu s \\ V_g &= 15 V \\ D &= 0.25 \\ R_l &= 150 \Omega \end{aligned} \quad (4.40)$$

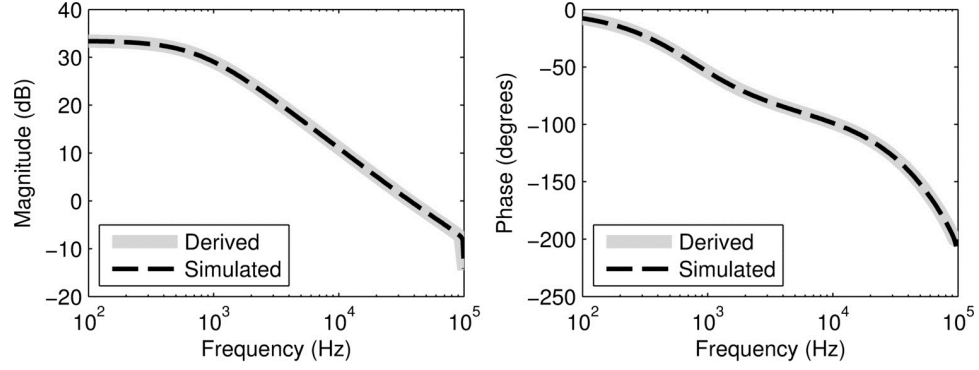


Figure 17: The derived NVM boost converter in DCM control-to-output transfer function (solid) compared with results obtained from simulation (dashed).

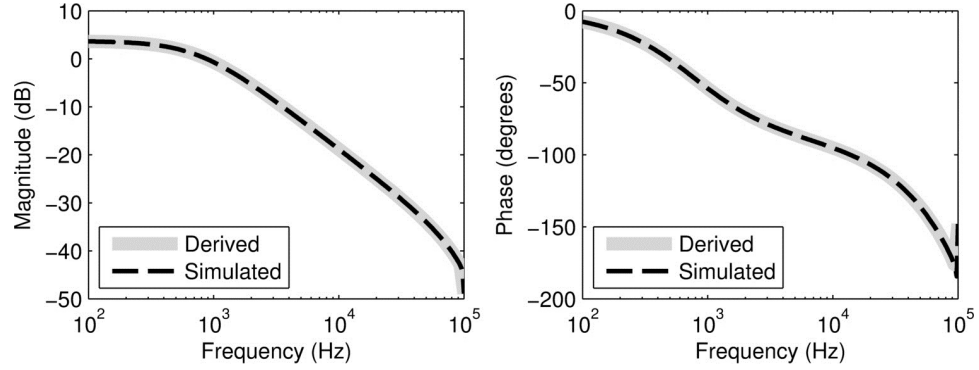


Figure 18: The derived NVM boost converter in DCM input-to-output transfer function (solid) compared with results obtained from simulation (dashed).

### Section 4.3: Normal Voltage Mode Converter in DVM

A power converter operates in DVM when the voltage  $v_1(t)$  across the storage capacitor ( $C_1$  in the OTB converter) goes to zero, causing the switching diode  $P$  to turn on prematurely and creating a new topology before  $r(t)$  dictates the shutting off of  $Q$ . As with DCM, DVM operation is generally undesirable and avoided through proper design as it makes  $v_{out}(t)$  dependent on  $R_l$ . With  $N_s = 3$ ,  $T_s$  is split into three subintervals with DC lengths  $D_1T_s$ ,  $D_2T_s$ , and  $D_3T_s$ . The effective topologies of the OTB converter's power stage during the three subintervals are shown in Fig. 19. Fig. 20 illustrates the switching period  $T_s$  for DVM described by both Model 1 and Model 2. Model 2 will now be considered for the derivations, after which the subscripts of the final result will be rotated so that it is applicable to Model 1. Notice that Model 2 starts two subintervals ahead of Model 1 in this case so that the  $r(t)$  affects both  $t_1$  and  $t_4$ .

$t_2$  is determined by the period  $T_s$  of the sawtooth signal so  $\hat{t}_2 = 0$ .  $t_3$  is the time when the voltage across  $C_1$ ,  $v_1(t)$ , goes to zero:

$$v_1(t_3) = 0 \tag{4.41}$$

Defining a vector  $h^T$  such that:

$$h^T x(t) = v_1(t) \tag{4.42}$$

allows (4.41) to be expressed as:

$$h^T x(t_3) = 0 \tag{4.43}$$

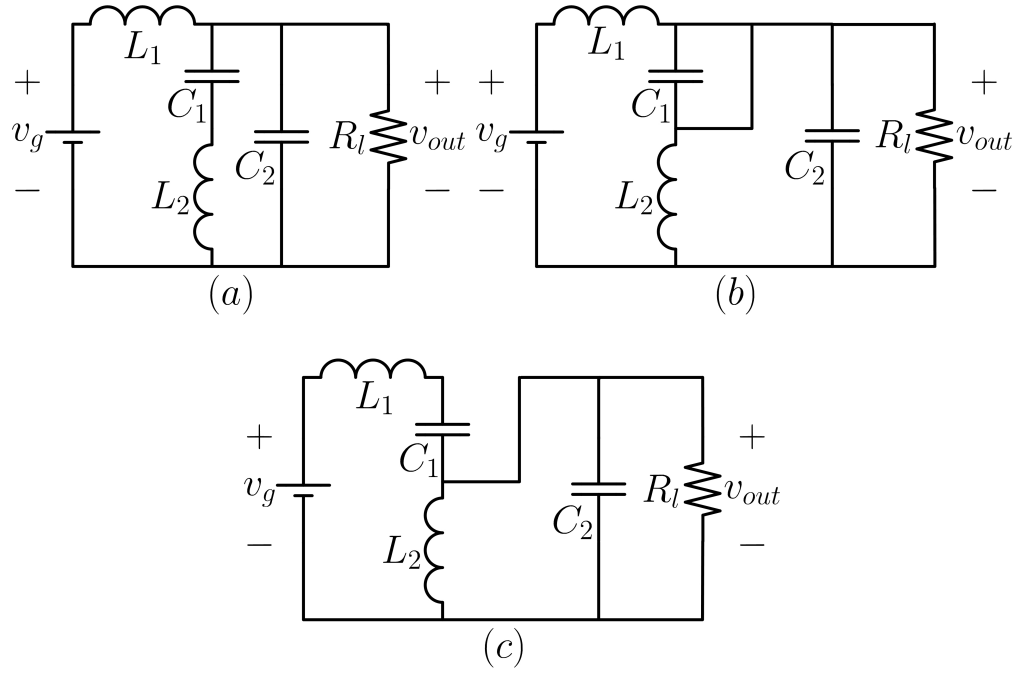


Figure 19: The effective topologies of a NVM OTB converter in DVM assuming ideal switches. (a) is the effective topology during the first subinterval of DC length  $D_1T_s$  where  $Q$  is on and  $P$  is off. (b) is the effective topology during the second subinterval of DC length  $D_2T_s$  where both  $Q$  and  $P$  are on. (c) is the effective topology during the third subinterval of DC length  $D_3T_s$  where  $Q$  is off and  $P$  is on.

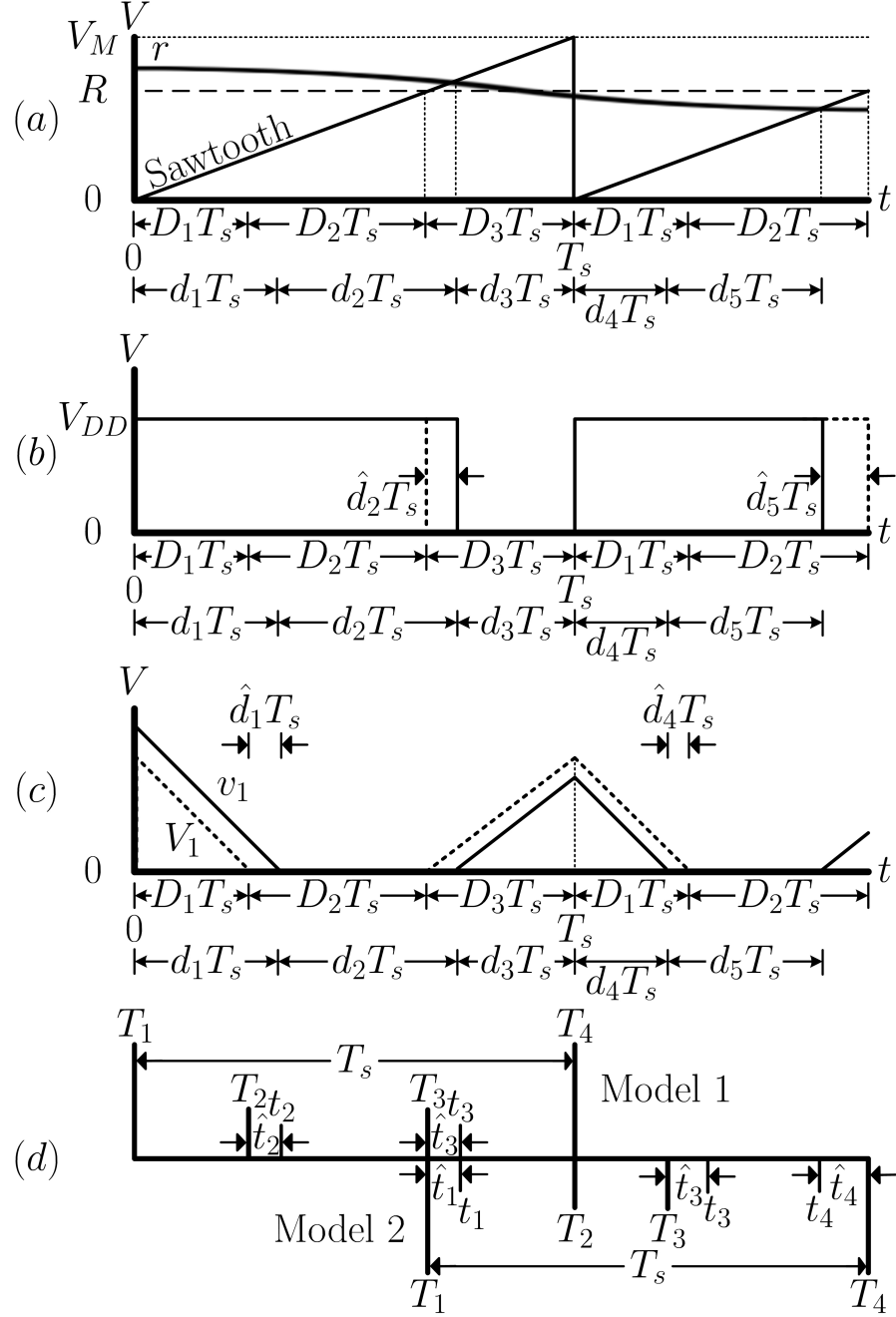


Figure 20: Switching period  $T_s$  for a NVM converter in DVM, described for both Model 1 and Model 2. (a) shows the sawtooth voltage of period  $T_s$  being compared to the control signal  $r(t)$  to produce the square wave (b). (c) shows the storage capacitor voltage  $v_1(t)$  reaching zero and creating a third intermediate topology. (d) illustrates how the times are denoted under Model 1 and Model 2.

Dropping the DC component of  $x(t_3)$ :

$$h^T \hat{x}(t_3) = 0 \quad (4.44)$$

Substituting in (2.19) and solving for  $\hat{t}_3$ :

$$\hat{t}_3 = -\mu_2 h^T \Phi_2 \hat{x}(T_2) - \mu_2 h^T \eta_2 B_2 \hat{u}(T_3) \quad (4.45)$$

where

$$\mu_2 = \frac{1}{h^T \gamma_2} \quad (4.46)$$

The expression for  $\hat{t}_4$  is the same as the expression for the  $\hat{t}_4$  of the DCM case:

$$\hat{t}_4 = \frac{T_s \hat{r}(T_4)}{V_M} = \mu_3 \hat{r}(T_4) \quad (4.47)$$

where

$$\mu_3 = \frac{T_s}{V_M} \quad (4.48)$$

Finding an expression for  $\hat{x}(T_1)$  is the next step after acquiring one for  $\hat{t}_4$ . By using (2.27),  $\hat{x}(T_2)$  is:

$$\hat{x}(T_2) = \Phi_1 \hat{x}(T_1) + \eta_1 B_1 \hat{u}(T_2) \quad (4.49)$$

Using (2.27) again and substituting in (4.45) and (4.49) gives the following expression for  $\hat{x}(T_3)$ :

$$\begin{aligned}
\hat{x}(T_3) &= \Phi_2 \hat{x}(T_2) + \xi_2 \hat{t}_3 + \eta_2 B_2 \hat{u}(T_3) \\
&= \Gamma_2 \Phi_2 \hat{x}(T_2) + \Gamma_2 \eta_2 B_2 \hat{u}(T_3) \\
&= \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}(T_2) + \Gamma_2 \eta_2 B_2 \hat{u}(T_3)
\end{aligned} \tag{4.50}$$

where

$$\Gamma_2 = I - \xi_2 \mu_2 h^T \tag{4.51}$$

Using (2.27) once more to find  $\hat{x}(T_3)$  and substituting in (4.47) and (4.50):

$$\begin{aligned}
\hat{x}(T_4) &= \Phi_3 \hat{x}(T_3) + \xi_3 \hat{t}_4 + \eta_3 B_3 \hat{u}(T_4) \\
&= \Phi_3 \hat{x}(T_3) + \xi_3 \mu_3 \hat{r}(T_4) + \eta_3 B_3 \hat{u}(T_4) \\
&= \Phi_3 \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}(T_4) + \Phi_3 \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}(T_2) \\
&\quad + \Phi_3 \Gamma_2 \eta_2 B_2 \hat{u}(T_3) + \eta_3 B_3 \hat{u}(T_4)
\end{aligned} \tag{4.52}$$

Expressing all of the signals in their exponential form as in (2.31) and substituting in (2.32) for  $\hat{x}(T_4)$ :

$$\begin{aligned}
\hat{x}(T_1) e^{j\omega T_s} &= \Phi_3 \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}_p e^{j\omega T_4} + \Phi_3 \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}_p e^{j\omega T_2} \\
&\quad + \Phi_3 \Gamma_2 \eta_2 B_2 \hat{u}_p e^{j\omega T_3} + \eta_3 B_3 \hat{u}_p e^{j\omega T_4}
\end{aligned} \tag{4.53}$$

Solving for  $\hat{x}(T_1)$ :

$$\begin{aligned}
\hat{x}(T_1) &= (I - \beta_3 \Gamma_2 \beta_2 \beta_1)^{-1} \{ \xi_3 \mu_3 \hat{r}(T_1) + [\beta_3 \Gamma_2 (\beta_2 \eta_1 B_1 + \eta_2 B_2) \\
&\quad + \eta_3 B_3] \hat{u}(T_1) \}
\end{aligned} \tag{4.54}$$

Now what is left is to substitute (4.47), (4.49), (4.50), and (4.54) into (3.23).



Expanding (3.23) for  $N_s = 3$ :

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} [C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \rho_1 \hat{u}_p + C_2^T \eta_2 \hat{x}(T_2) e^{-j\omega T_2} + \zeta_2 \hat{t}_3 e^{-j\omega T_3} \\
& + \rho_2 \hat{u}_p + C_3^T \eta_3 \hat{x}(T_3) e^{-j\omega T_3} + \zeta_3 \hat{t}_4 e^{-j\omega T_4} + \rho_3 \hat{u}_p] \quad (4.55)
\end{aligned}$$

Substituting in (4.45), (4.47), (4.49), and (4.50) gives:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left( \{C_1^T \eta_1 + [C_2^T \eta_2 + (C_3^T \eta_3 \Gamma_2 - \zeta_2 \mu_2 h^T) \beta_2] \beta_1\} \hat{x}(T_1) e^{-j\omega T_1} \right. \\
& + \zeta_3 \mu_3 \hat{r}_p + \{[C_2^T \eta_2 + (C_3^T \eta_3 \Gamma_2 - \zeta_2 \mu_2 h^T) \beta_2] \eta_1 B_1 \\
& \left. + (C_3^T \eta_3 \Gamma_2 - \zeta_2 \mu_2 h^T) \eta_2 B_2 + \rho_1 + \rho_2 + \rho_3\} \hat{u}_p \right) \quad (4.56)
\end{aligned}$$

Finally substituting in (4.54):

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left[ \left( \{C_1^T \eta_1 + [C_2^T \eta_2 + (C_3^T \eta_3 \Gamma_2 - \zeta_2 \mu_2 h^T) \beta_2] \beta_1\} \right. \right. \\
& \times (I - \beta_3 \Gamma_2 \beta_2 \beta_1)^{-1} \xi_3 \mu_3 + \zeta_3 \mu_3 \Big) \hat{r}_p + \left( \{C_1^T \eta_1 + [C_2^T \eta_2 \right. \\
& + (C_3^T \eta_3 \Gamma_2 - \zeta_2 \mu_2 h^T) \beta_2] \beta_1\} (I - \beta_3 \Gamma_2 \beta_2 \beta_1)^{-1} [\beta_3 \Gamma_2 (\beta_2 \eta_1 B_1 \\
& + \eta_2 B_2) + \eta_3 B_3] + [C_2^T \eta_2 + (C_3^T \eta_3 \Gamma_2 - \zeta_2 \mu_2 h^T) \beta_2] \eta_1 B_1 \\
& \left. \left. + (C_3^T \eta_3 \Gamma_2 - \zeta_2 \mu_2 h^T) \eta_2 B_2 + \rho_1 + \rho_2 + \rho_3\} \hat{u}_p \right) \right] \quad (4.57)
\end{aligned}$$

In DVM, rotating the subscripts back to Model 1 requires two rotations forwards (or one rotation backwards) such that 1 is replaced with 3, 2 with 1, and 3 with 2:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left[ \left( \{C_3^T \eta_3 + [C_1^T \eta_1 + (C_2^T \eta_2 \Gamma_1 - \zeta_1 \mu_1 h^T) \beta_1] \beta_3\} \right. \right. \\
& \times (I - \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} \xi_2 \mu_2 + \zeta_2 \mu_2 \Big) \hat{r}_p + \left( \{C_3^T \eta_3 + [C_1^T \eta_1 \right. \\
& + (C_2^T \eta_2 \Gamma_1 - \zeta_1 \mu_1 h^T) \beta_1] \beta_3\} (I - \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} [\beta_2 \Gamma_1 (\beta_1 \eta_3 B_3 \\
& + \eta_1 B_1) + \eta_2 B_2] + [C_1^T \eta_1 + (C_2^T \eta_2 \Gamma_1 - \zeta_1 \mu_1 h^T) \beta_1] \eta_3 B_3 \\
& \left. \left. + (C_2^T \eta_2 \Gamma_1 - \zeta_1 \mu_1 h^T) \eta_1 B_1 + \rho_1 + \rho_2 + \rho_3 \right) \hat{u}_p \right] \quad (4.58)
\end{aligned}$$

where

$$\begin{aligned}
\mu_1 &= \frac{1}{h^T \gamma_1} \\
\mu_2 &= \frac{T_s}{V_M}
\end{aligned} \quad (4.59)$$

By setting  $\hat{r}_p = 1$  and  $\hat{u}_p = 0$  the control-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{r}}$  can be found:

$$\begin{aligned}
\frac{\hat{v}_{out}}{\hat{r}} = & \frac{1}{V_M} \left( \{C_3^T \eta_3 + [C_1^T \eta_1 + (C_2^T \eta_2 \Gamma_1 - \zeta_1 \mu_1 h^T) \beta_1] \beta_3\} \right. \\
& \times (I - \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} \xi_2 + \zeta_2 \Big) \quad (4.60)
\end{aligned}$$

Setting  $\hat{r}_p = 0$  and  $\hat{u}_p = 1$ , the input-to-output transfer function  $\frac{v_{out}}{\hat{v}_g}$  is:

$$\begin{aligned}
\frac{\hat{v}_{out}}{\hat{v}_g} = & \frac{1}{T_s} \left( \{C_3^T \eta_3 + [C_1^T \eta_1 + (C_2^T \eta_2 \Gamma_1 - \zeta_1 \mu_1 h^T) \beta_1] \beta_3\} \right. \\
& \times (I - \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} [\beta_2 \Gamma_1 (\beta_1 \eta_3 B_3 + \eta_1 B_1) + \eta_2 B_2] \\
& + [C_1^T \eta_1 + (C_2^T \eta_2 \Gamma_1 - \zeta_1 \mu_1 h^T) \beta_1] \eta_3 B_3 + (C_2^T \eta_2 \Gamma_1 \\
& \left. - \zeta_1 \mu_1 h^T) \eta_1 B_1 + \rho_1 + \rho_2 + \rho_3 \Big) \quad (4.61)
\end{aligned}$$

The control-to-output and input-to-output transfer functions are compared with results from simulation in Fig. 21 and Fig. 22, respectively, with the

following parameters:

$$\begin{aligned}T_s &= 10\mu s \\V_g &= 28V \\D &= 0.4 \\R_l &= 3\Omega\end{aligned}\tag{4.62}$$

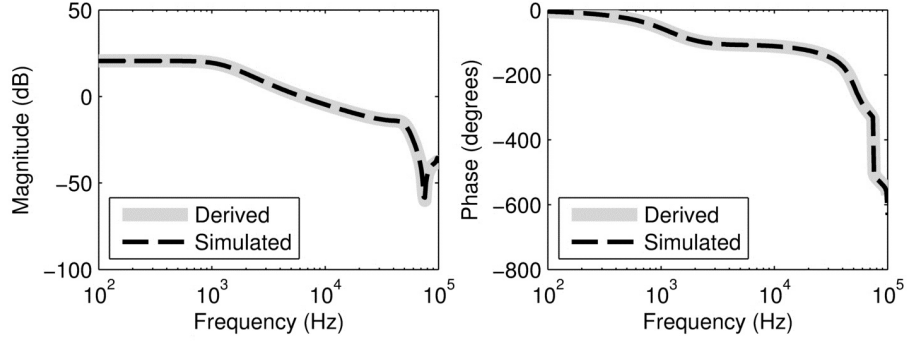


Figure 21: The derived NVM boost converter in DVM control-to-output transfer function (solid) compared with results obtained from simulation (dashed).

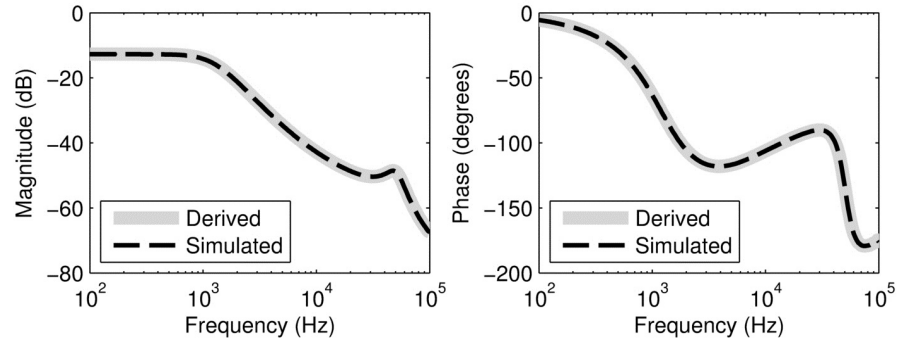


Figure 22: The derived NVM boost converter in DVM input-to-output transfer function (solid) compared with results obtained from simulation (dashed).

## Chapter 5: Current Programmed Transfer Functions

A CPM power converter uses the current through the active switch  $i_Q(t)$  as the sawtooth signal, allowing  $r(t)$  to limit the active switch current which in turn lowers output voltage ripple. Controlling the peak switch current also allows power converters in parallel sharing the same  $r(t)$  to have equal peak currents so that the power is distributed equally. As shown in Fig. 23 and Fig. 24,  $i_Q(t)$  is sensed through a sensing resistor  $R_s$  and compared to  $r(t)$ . The output voltage controllers have been omitted in Fig. 23 and Fig. 24 since the open loop transfer functions are derived.  $r(t)$  contains the peak active switch current  $i_{max}(t)$  the DC value of which,  $I_{max}$ , is related to  $V_{ref}$  by:

$$V_{ref} = I_{max} R_s \quad (5.1)$$

Because  $i_Q(t)$  will drop as soon as  $i_{max}(t)$  is reached and the topology switches, the output of the comparator is fed into the reset input R of a SR-latch in order to turn off the active switch  $Q$  and prevent  $Q$  from turning back on until a clock signal CLK, which has an impulse every  $T_s$  seconds and is fed into the set input S of the SR-latch, turns the active switch  $Q$  back on. The transfer functions of interest will be derived for the CPM boost converter in Fig. 23 under CCM and DCM, followed by the CPM OTB converter in Fig. 24 under DVM. It should be noted that the exact  $\frac{v_{out}}{\hat{r}}$  for a CPM converter operating in CCM and DCM has already been derived in [2], however  $\frac{v_{out}}{\hat{u}}$  for a CPM converter in CCM and DCM along with the exact transfer functions associated with a CPM converter in DVM has not yet been published.

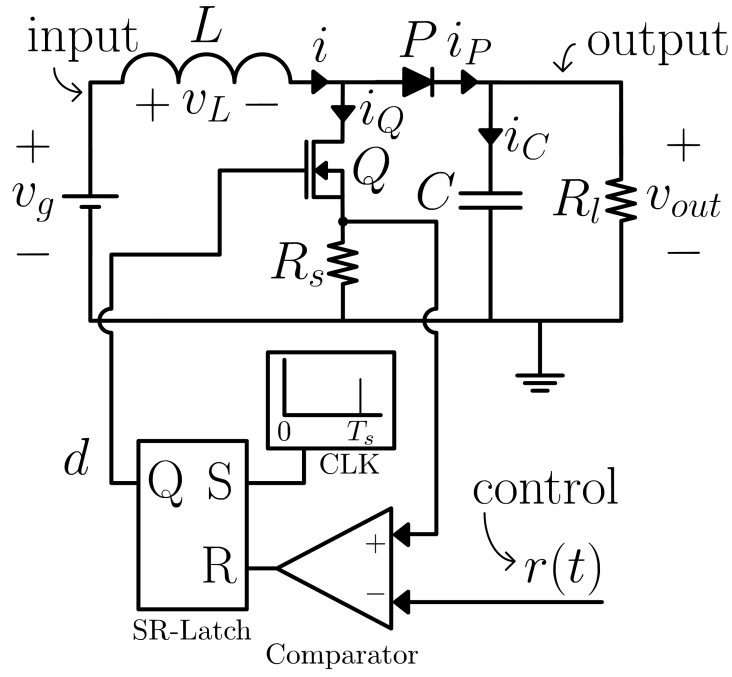


Figure 23: Current programmed boost converter schematic. The current  $i_Q(t)$  is compared to a maximum current  $i_{max}(t)$ , which is contained in the signal  $r(t)$ .  $i_Q(t)$  is sensed through a sensing resistor  $R_s$  and compared to the control signal  $r(t)$ . An SR-Latch and a clock signal CLK are implemented such that a square wave is produced with a period  $T_s$  and duty ratio  $d$ .

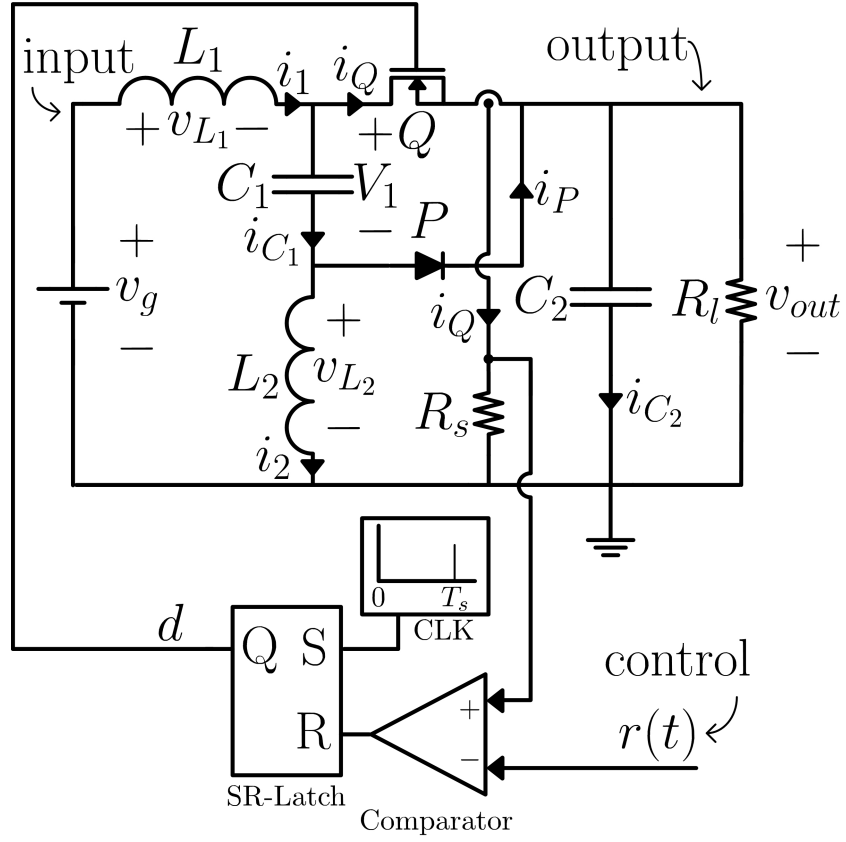


Figure 24: Current programmed OTB converter schematic. The current  $i_Q(t)$  is compared to a maximum current  $i_{max}(t)$ , which is contained in the signal  $r(t)$ .  $i_Q(t)$  is sensed through a sensing resistor  $R_s$  and compared to the control signal  $r(t)$ . An SR-Latch and a clock signal CLK are implemented such that a square wave is produced with a period  $T_s$  and duty ratio  $d$ .

### Section 5.1: Current Programmed Converter in CCM

A CPM converter in CCM has only two topologies ( $N_s = 2$ ) and therefore has two subintervals over the period  $T_s$ . The first subinterval will have a DC length  $DT_s$  and the second subinterval will be of DC length  $(1 - D)T_s$ . The effective topologies of the boost power stage during the two subintervals are shown in Fig. 25. For Model 1, which will first be considered for clarity,  $DT_s$  is equal the time it takes for  $I_Q(t)$  to reach  $I_{max}(t)$  where  $I_Q(t)$  and  $I_{max}(t)$  are the DC components of  $i_Q(t)$  and  $i_{max}(t)$ , respectively. As illustrated in Fig. 26a, an  $i_{max}(t)$  that results in  $D > 0.5$  can cause the system to be unstable. For this reason, a stability ramp of slope  $m$  may be added to  $i_Q(t)$  when compared to  $I_{max}$ . A more convenient way of considering the added slope is to subtract the slope from  $i_{max}(t)$  as shown in Fig. 26b which can handle  $D > 0.5$  by reducing any deviation from the DC average over time. Fig. 27 shows the period  $T_s$  for both Model 1 and Model 2. Model 2 will now be considered for the following derivations until the subscripts of the final result are rotated back to Model 1.

$t_2$  is determined by the CLK signal and is not sensitive to variations in  $r(t)$  or  $u(t)$ , therefore  $\hat{t}_2 = 0$ . Finding an expression for  $\hat{t}_3$  is accomplished by examining the switch condition which is the time when the  $r(t)$  is equal to a linear combination of the states  $h^T x(t_3)$  with an added stability ramp of slope  $m$ :

$$r(t_3) = h^T x(t_3) + mt_3 \quad (5.2)$$

where  $h^T$  is a vector such that:

$$i_Q(t) = h^T x(t) \quad (5.3)$$



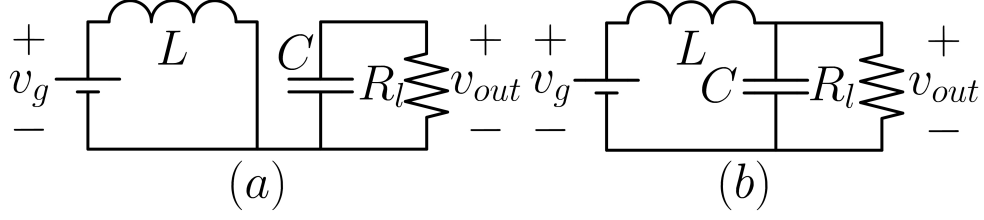


Figure 25: The effective topologies of a CPM boost converter in CCM assuming ideal switches. (a) is the effective topology during the first subinterval of DC length  $DT_s$  where  $Q$  is on and  $P$  is off. (b) is the effective topology during the second subinterval of DC length  $(1 - D)T_s$  where  $Q$  is off and  $P$  is on.

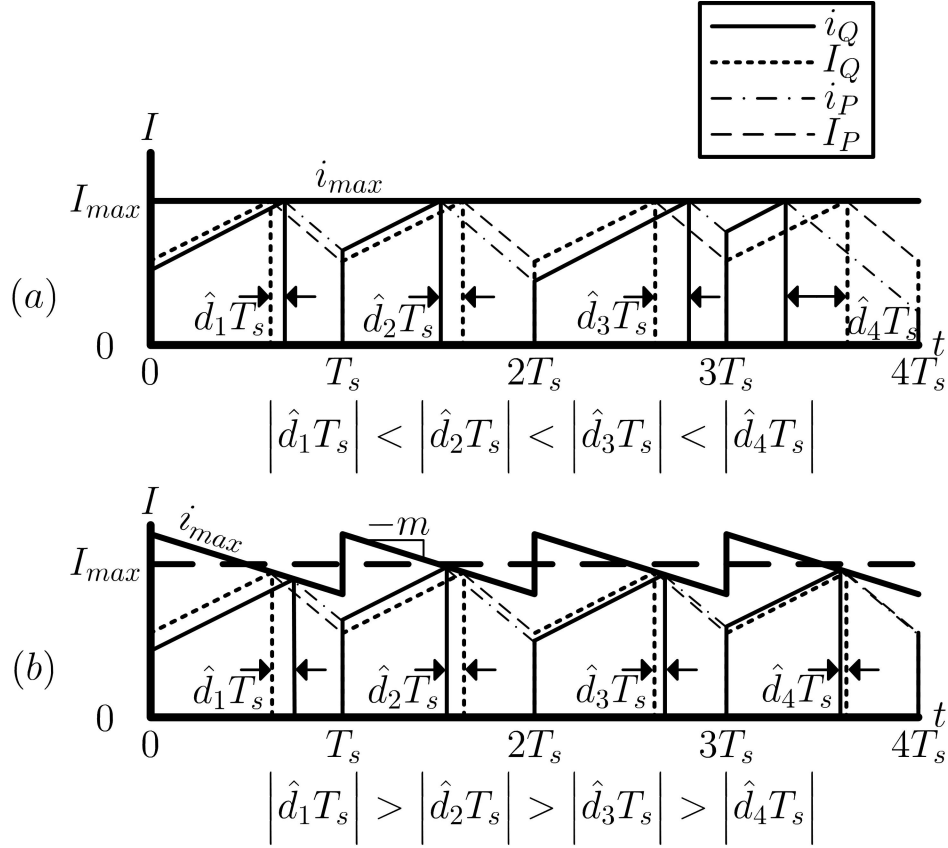


Figure 26: (a) shows a the potential instability for a CPM converter with  $D > 0.5$  by showing the increase in the magnitudes of  $d_i$  over multiple switching periods  $T_s$ . (b) shows that an added stability ramp of slope  $m$  makes the current settle back to its average (DC) signal as time progresses, such that  $d_i$  decreases for increasing  $i$ .

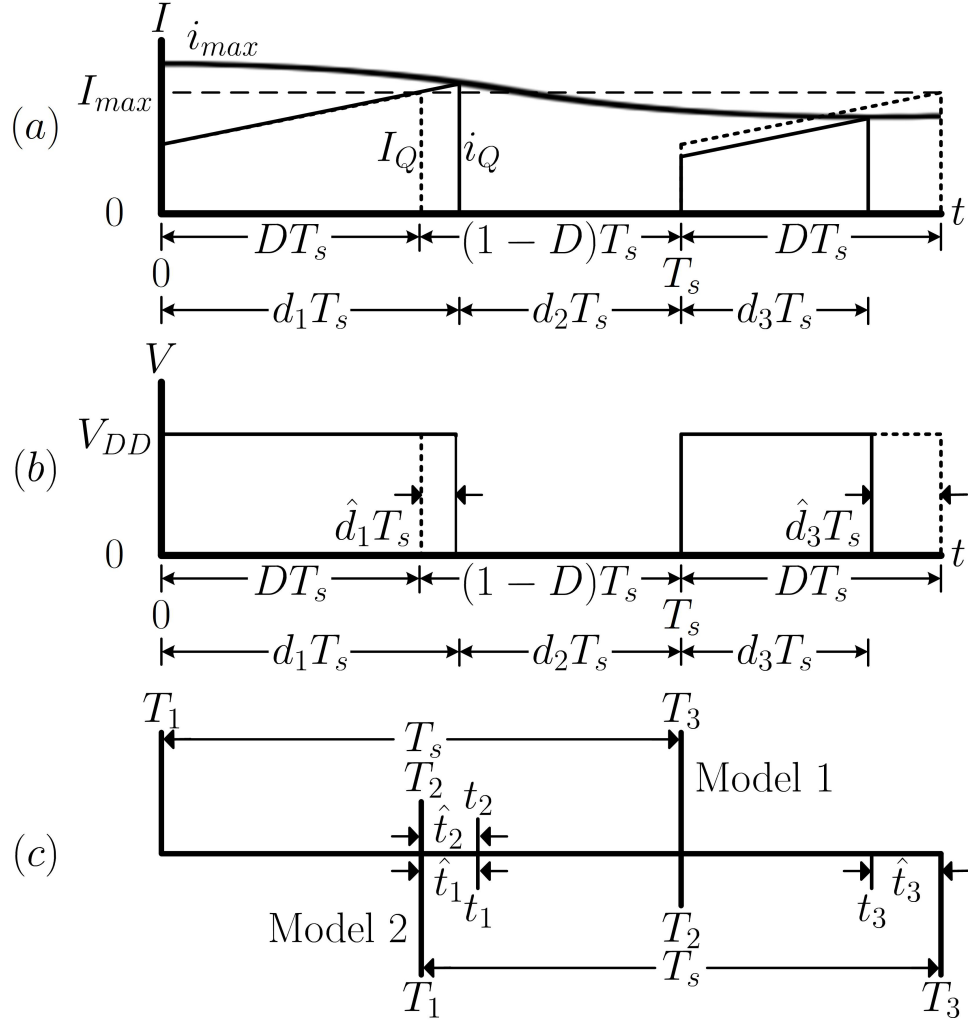


Figure 27: Switching period  $T_s$  for a CPM converter in CCM, described for both Model 1 and Model 2. (a) shows the active switch current  $i_Q$  being compared to the peak active switch current  $i_{max}(t)$  to produce the square wave (b). (c) illustrates how the times are denoted under Model 1 and Model 2.

Dropping all DC terms of (5.2) gives:

$$\hat{r}(T_3) = h^T \hat{x}(t_3) + m \hat{t}_3 \quad (5.4)$$

Substituting in (2.19) for  $\hat{x}(t_3)$  and solving for  $\hat{t}_3$  results in:

$$\hat{t}_3 = \mu \hat{r}(T_3) - \mu h^T \Phi_2 \hat{x}(T_2) - \mu h^T \eta_2 B_2 \hat{u}(T_3) \quad (5.5)$$

where

$$\mu = \frac{1}{h^T \gamma_2 + m} \quad (5.6)$$

With an expression for  $\hat{t}_3$  the next step is to get an expression for  $\hat{x}(T_1)$ . This is done by solving the difference equation (2.27) by going through one switching period  $T_s$ .  $\hat{x}(T_2)$  is given by:

$$\hat{x}(T_2) = \Phi_1 \hat{x}(T_1) + \eta_1 B_1 \hat{u}(T_2) \quad (5.7)$$

Using (2.27) once again to find  $\hat{x}(T_3)$  and substituting in (5.7) and (5.5) gives:

$$\begin{aligned} \hat{x}(T_3) &= \Phi_2 \hat{x}(T_2) + \xi_2 \hat{t}_3 + \eta_2 B_2 \hat{u}(T_3) \\ &= \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \xi_2 \mu \hat{r}(T_3) + \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}(T_2) + \Gamma_2 \eta_2 B_2 \hat{u}(T_3) \end{aligned} \quad (5.8)$$

where

$$\Gamma_2 = I - \xi_2 \mu h^T \quad (5.9)$$

Expressing all of the signals in their exponential form as in (2.31) and substituting in (2.32) for  $\hat{x}(T_3)$ :

$$\hat{x}(T_1)e^{j\omega T_s} = \Gamma_2\Phi_2\Phi_1\hat{x}(T_1) + \xi_2\mu\hat{r}_p e^{j\omega T_4} + \Gamma_2\Phi_2\eta_1 B_1\hat{u}_p e^{j\omega T_2} + \Gamma_2\eta_2 B_2\hat{u}_p e^{j\omega T_4} \quad (5.10)$$

Solving for  $\hat{x}(T_1)$ :

$$\hat{x}(T_1) = (I - \Gamma_2\beta_2\beta_1)^{-1}[\xi_2\mu\hat{r}(T_1) + \Gamma_2(\beta_2\eta_1 B_1 + \eta_2 B_2)\hat{u}(T_1)] \quad (5.11)$$

The last step is to substitute (5.5), (5.7) and (5.11) into (3.23). Expanding (3.23) results in:

$$\begin{aligned} H_0(j\omega) = \frac{1}{T_s} [C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \rho_1 \hat{u}_p + C_2^T \eta_2 \hat{x}(T_2) e^{-j\omega T_2} \\ + \zeta_2 \hat{t}_3 e^{-j\omega T_3} + \rho_2 \hat{u}_p] \end{aligned} \quad (5.12)$$

Substituting in (5.5):

$$\begin{aligned} H_0(j\omega) = \frac{1}{T_s} [C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \zeta_2 \mu \hat{r}_p + (C_2^T \eta_2 \\ - \zeta_2 \mu h^T \beta_2) \hat{x}(T_2) e^{-j\omega T_2} + (\rho_1 + \rho_2 - \zeta_2 \mu h^T \eta_2 B_2) \hat{u}_p] \end{aligned} \quad (5.13)$$

Now substituting in (5.7):

$$\begin{aligned} H_0(j\omega) = \frac{1}{T_s} \{ [C_1^T \eta_1 + (C_2^T \eta_2 - \zeta_2 \mu h^T \beta_2) \beta_1] \hat{x}(T_1) e^{-j\omega T_1} + \zeta_2 \mu \hat{r}_p \\ + [(C_2^T \eta_2 - \zeta_2 \mu h^T \beta_2) \eta_1 B_1 - \zeta_2 \mu h^T \eta_2 B_2 + \rho_1 + \rho_2] \hat{u}_p \} \end{aligned} \quad (5.14)$$

Finally substituting in (5.11):

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left( \{[C_1^T \eta_1 + (C_2^T \eta_2 - \zeta_2 \mu h^T \beta_2) \beta_1](I - \Gamma_2 \beta_2 \beta_1)^{-1} \xi_2 \mu \right. \\
& + \zeta_2 \mu \} \hat{r}_p + \{[C_1^T \eta_1 + (C_2^T \eta_2 - \zeta_2 \mu h^T \beta_2) \beta_1](I - \Gamma_2 \beta_2 \beta_1)^{-1} \\
& \times [\Gamma_2 (\beta_2 \eta_1 B_1 + \eta_2 B_2)] + (C_2^T \eta_2 - \zeta_2 \mu h^T \beta_2) \eta_1 B_1 \\
& \left. - \zeta_2 \mu h^T \eta_2 B_2 + \rho_1 + \rho_2 \} \hat{u}_p \right) \quad (5.15)
\end{aligned}$$

Rotating the subscripts so that the final result is in Model 1:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left( \{[C_2^T \eta_2 + (C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \beta_2](I - \Gamma_1 \beta_1 \beta_2)^{-1} \xi_1 \mu \right. \\
& + \zeta_1 \mu \} \hat{r}_p + \{[C_2^T \eta_2 + (C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \beta_2](I - \Gamma_1 \beta_1 \beta_2)^{-1} \\
& \times [\Gamma_1 (\beta_1 \eta_2 B_2 + \eta_1 B_1)] + (C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \eta_2 B_2 \\
& \left. - \zeta_1 \mu h^T \eta_1 B_1 + \rho_1 + \rho_2 \} \hat{u}_p \right) \quad (5.16)
\end{aligned}$$

where

$$\mu = \frac{1}{h^T \gamma_1 + m} \quad (5.17)$$

As with the open loop derivations, the transfer functions are found by setting the perturbation of the input of interest to one while setting the other to zero. The control-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{r}}$  is found by setting  $\hat{r}_p = 1$  and setting the  $\hat{u}_p = 0$ :

$$\frac{\hat{v}_{out}}{\hat{r}} = \frac{\mu}{T_s} \{[(C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \beta_2 + C_2^T \eta_2](I - \Gamma_1 \beta_1 \beta_2)^{-1} \xi_1 + \zeta_1\} \quad (5.18)$$

The input-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{v}_g}$  is found by setting  $\hat{r}_p = 0$  and  $\hat{u}_p = 1$ :

$$\begin{aligned} \frac{\hat{v}_{out}}{\hat{v}_g} = & \frac{1}{T_s} \{ [(C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \beta_2 + C_2^T \eta_2] (I - \Gamma_1 \beta_1 \beta_2)^{-1} [\Gamma_1 (\eta_1 B_1 \\ & + \beta_1 \eta_2 B_2)] - \zeta_1 \mu h^T \eta_1 B_1 + (C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \eta_2 B_2 + \rho_1 + \rho_2 \} \quad (5.19) \end{aligned}$$

The control-to-output and input-to-output transfer functions are compared with results from simulation in Fig. 28 and Fig. 29, respectively, with the following parameters:

$$\begin{aligned} T_s &= 10 \mu s \\ V_g &= 15 V \\ D &= 0.25 \\ R_l &= 18.6 \Omega \\ R_s &= 0.5 \Omega \\ m &= 0 V/s \end{aligned} \quad (5.20)$$

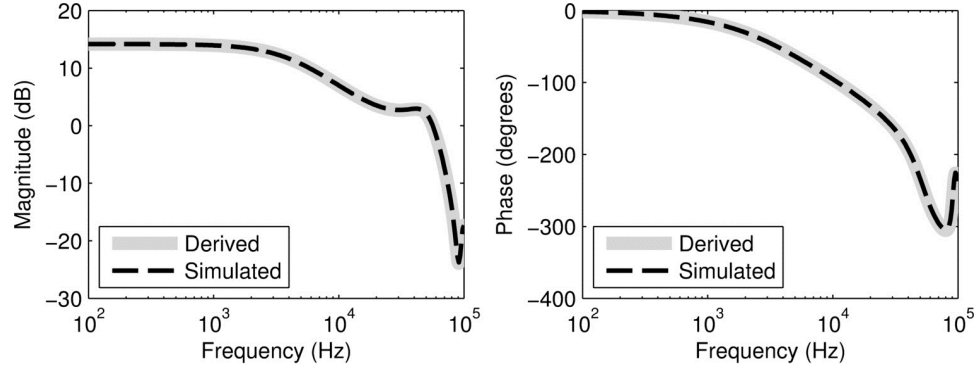


Figure 28: The derived current programmed boost converter in CCM control-to-output transfer function (solid) compared with results obtained from simulation (dashed).

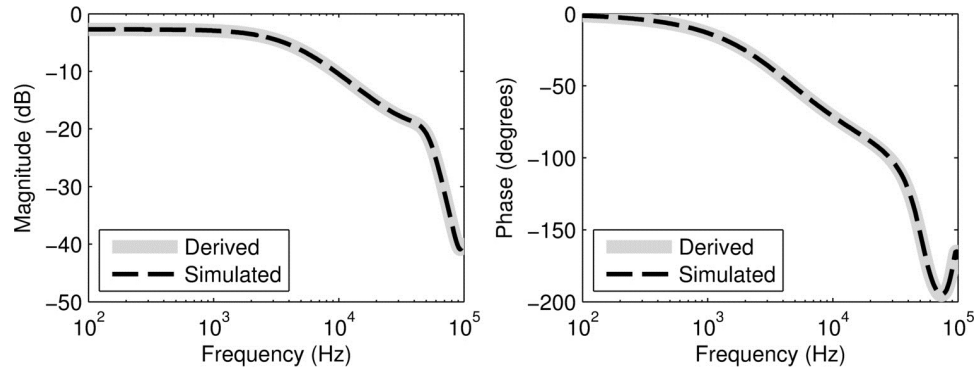


Figure 29: The derived current programmed boost converter in CCM input-to-output transfer function (solid) compared with results obtained from simulation (dashed).

## Section 5.2: Current Programmed Converter in DCM

As with a NVM converter, a current programmed power converter operates in DCM when  $i_P(t)$  goes to zero and the switching diode  $P$  shuts off before  $Q$  is turned back on. This results in a third topology such that  $N_s = 3$  and  $T_s$  is split into three subintervals of DC lengths  $D_1T_s$ ,  $D_2T_s$ , and  $D_3T_s$  shown by Fig. 30. The effective topologies of the boost power stage during the three subintervals are shown in Fig. 31. Model 2, as shown in Fig. 30, will be used for the following derivation, and then the subscripts of the final result will be rotated appropriately back to Model 1.

$\hat{t}_2$  will be found by the same procedure used in the open loop DCM case.  $t_2$  is when the switching diode current  $i_P(t)$  goes to zero:

$$i_P(t_2) = 0 \quad (5.21)$$

Defining a vector  $h_1^T$  such that:

$$h_1^T x(t) = i_P(t) \quad (5.22)$$

allows (5.21) to be expressed as:

$$h_1^T x(t_2) = 0 \quad (5.23)$$

Dropping the DC component of (5.23) gets:

$$h_1^T \hat{x}(t_2) = 0 \quad (5.24)$$

Substituting in (2.19) and solving for  $\hat{t}_2$ :



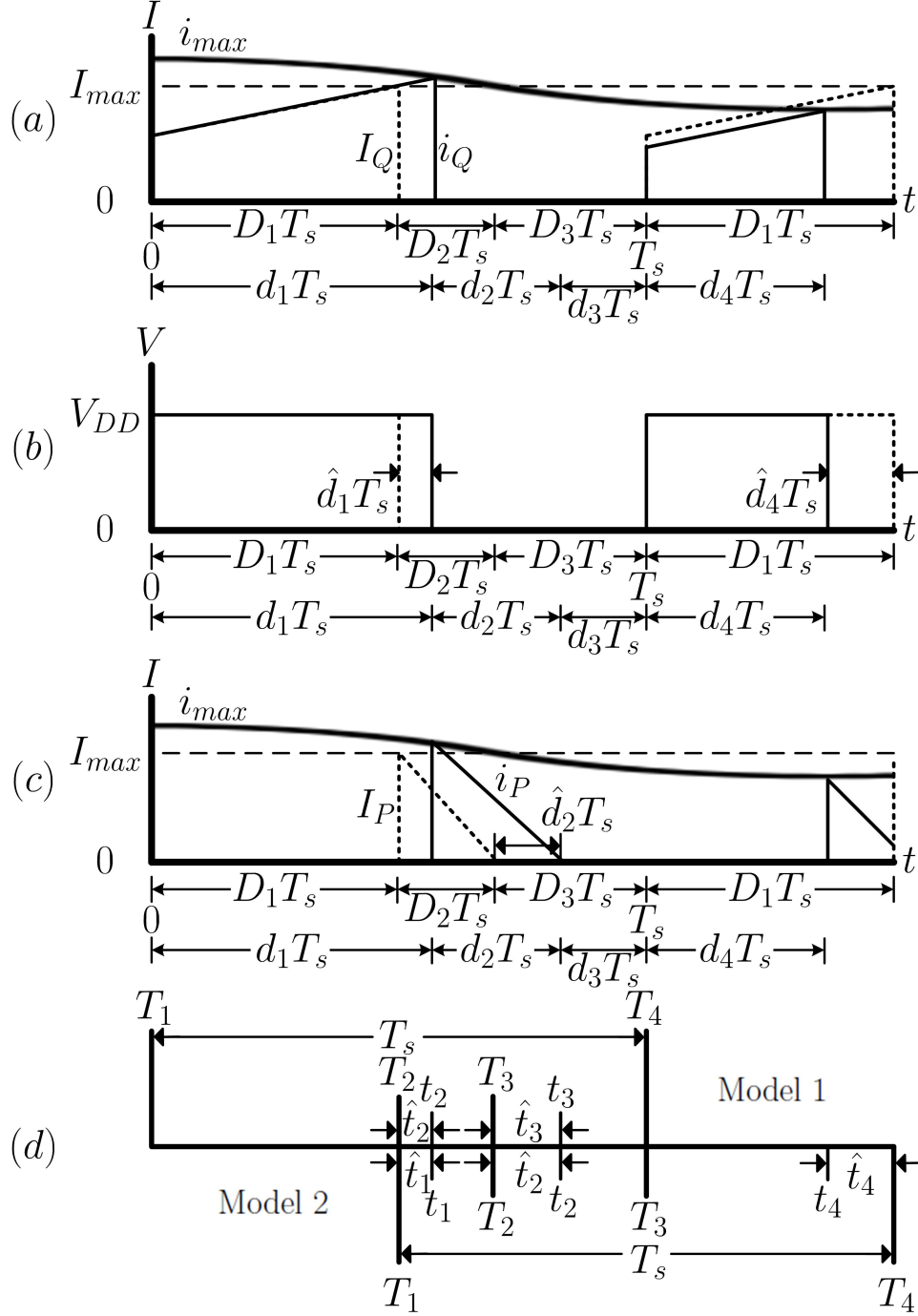


Figure 30: Switching period  $T_s$  for a CPM converter in DCM, described for both Model 1 and Model 2. (a) shows the current  $i_Q(t)$  being compared to a maximum current  $i_{max}(t)$  to produce the square wave (b). (c) shows the diode current  $i_P(t)$  reaching zero and creating a third topology. (d) illustrates how the times are denoted under Model 1 and Model 2.

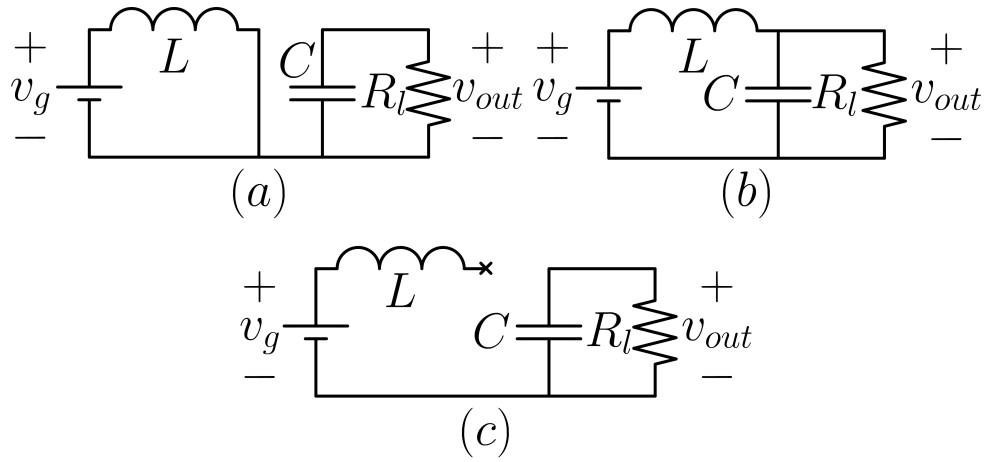


Figure 31: The effective topologies of a CPM boost converter in DCM assuming ideal switches. (a) is the effective topology during the first subinterval of DC length  $D_1T_s$  where  $Q$  is on and  $P$  is off. (b) is the effective topology during the second subinterval of DC length  $D_2T_s$  where  $Q$  is off and  $P$  is on. (c) is the effective topology during the third subinterval of DC length  $D_3T_s$  where both  $Q$  and  $P$  are off.

$$\hat{t}_2 = -\mu_1 h_1^T \Phi_1 \hat{x}(T_1) - \mu_1 h_1^T \eta_1 B_1 \hat{u}(T_2) \quad (5.25)$$

where

$$\mu_1 = \frac{1}{h_1^T \gamma_1} \quad (5.26)$$

$t_3$  is determined by the period  $T_s$  of the clock signal CLK and is not sensitive to variations in  $r(t)$  or  $u(t)$ , therefore  $\hat{t}_3 = 0$ . Finding  $\hat{t}_4$  involves the same procedure used for  $\hat{t}_3$  in the CPM converter operating in CCM case:

$$\hat{t}_4 = \mu_3 \hat{r}(T_4) - \mu_3 h_3^T \Phi_3 \hat{x}(T_3) - \mu_3 h_3^T \eta_3 B_3 \hat{u}(T_4) \quad (5.27)$$

where

$$\mu_3 = \frac{1}{h_3^T \gamma_3 + m} \quad (5.28)$$

$$h_3^T x(t) = i_Q(t) \quad (5.29)$$

and  $m$  is the slope of the added stability ramp.

$\hat{x}(T_1)$  will be found next. By using (2.27) and substituting in (5.25), an expression for  $\hat{x}(T_2)$  is found:

$$\begin{aligned} \hat{x}(T_2) &= \Phi_1 \hat{x}(T_1) + \xi_1 \hat{t}_2 + \eta_1 B_1 \hat{u}(T_2) \\ &= \Gamma \Phi_1 \hat{x}(T_1) + \Gamma_1 \eta_1 B_1 \hat{u}(T_2) \end{aligned} \quad (5.30)$$

where

$$\Gamma_i = I - \xi_i \mu_i h_i^T \quad (5.31)$$

Using (2.27) again and substituting in (5.30) gets an expression for  $\hat{x}(T_3)$ :

$$\begin{aligned}\hat{x}(T_3) &= \Phi_2 \hat{x}(T_2) + \eta_2 B_2 \hat{u}(T_3) \\ &= \Phi_2 \Gamma_1 \Phi_1 \hat{x}(T_1) + \Phi_2 \Gamma_1 \eta_1 B_1 \hat{u}(T_2) + \eta_2 B_2 \hat{u}(T_3)\end{aligned}\tag{5.32}$$

Using (2.27) once more and substituting in (5.27) and (5.32) gives an expression for  $\hat{x}(T_4)$ :

$$\begin{aligned}\hat{x}(T_4) &= \Phi_3 \hat{x}(T_3) + \xi_3 \hat{t}_4 + \eta_3 B_3 \hat{u}(T_4) \\ &= \Gamma_3 \Phi_3 \hat{x}(T_3) + \xi_3 \mu_3 \hat{r}(T_4) + \Gamma_3 \eta_3 B_3 \hat{u}(T_4) \\ &= \Gamma_3 \Phi_3 \Phi_2 \Gamma_1 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}(T_4) + \Gamma_3 \Phi_3 \Phi_2 \Gamma_1 \eta_1 B_1 \hat{u}(T_2) \\ &\quad + \Gamma_3 \Phi_3 \eta_2 B_2 \hat{u}(T_3) + \Gamma_3 \eta_3 B_3 \hat{u}(T_4)\end{aligned}\tag{5.33}$$

Expressing all of the signals in their exponential form as in (2.31) and substituting in (2.32) for  $\hat{x}(T_4)$ :

$$\begin{aligned}\hat{x}(T_1) e^{j\omega T_s} &= \Gamma_3 \Phi_3 \Phi_2 \Gamma_1 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}_p e^{j\omega T_4} + \Gamma_3 \Phi_3 \Phi_2 \Gamma_1 \eta_1 B_1 \hat{u}_p e^{j\omega T_2} \\ &\quad + \Gamma_3 \Phi_3 \eta_2 B_2 \hat{u}_p e^{j\omega T_3} + \Gamma_3 \eta_3 B_3 \hat{u}_p e^{j\omega T_4}\end{aligned}\tag{5.34}$$

Solving for  $\hat{x}(T_1)$ :

$$\begin{aligned}\hat{x}(T_1) &= (I - \Gamma_3 \beta_3 \beta_2 \Gamma_1 \beta_1)^{-1} \{ \xi_3 \mu_3 \hat{r}(T_1) + \Gamma_3 [\beta_3 (\beta_2 \Gamma_1 \eta_1 B_1 \\ &\quad + \eta_2 B_2) + \eta_3 B_3] \hat{u}(T_1) \}\end{aligned}\tag{5.35}$$

The final step is to substitute (5.27), (5.30), (5.32), and (5.35) into (3.23). Expanding (3.23) for  $N_s = 3$ :

$$\begin{aligned} H_0(j\omega) = & \frac{1}{T_s} [C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \zeta_1 \hat{t}_2 e^{-j\omega T_2} + \rho_1 \hat{u}_p + C_2^T \eta_2 \hat{x}(T_2) e^{-j\omega T_2} \\ & + \rho_2 \hat{u}_p + C_3^T \eta_3 \hat{x}(T_3) e^{-j\omega T_3} + \zeta_3 \hat{t}_4 e^{-j\omega T_4} + \rho_3 \hat{u}_p] \quad (5.36) \end{aligned}$$

Substituting in (5.25), (5.27), (5.30), and (5.32):

$$\begin{aligned} H_0(j\omega) = & \frac{1}{T_s} \left[ \left( C_1^T \eta_1 + \{ [C_2^T \eta_2 + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \beta_2] \Gamma_1 - \zeta_1 \mu_1 h_1^T \} \beta_1 \right) \right. \\ & \times \hat{x}(T_1) e^{-j\omega T_1} + \zeta_3 \mu_3 \hat{r}_p + \left( \{ [C_2^T \eta_2 + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \beta_2] \Gamma_1 \right. \\ & - \zeta_1 \mu_1 h_1^T \} \eta_1 B_1 + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \eta_2 B_2 - \zeta_3 \mu_3 h_3^T \eta_3 B_3 \\ & \left. \left. + \rho_1 + \rho_2 + \rho_3 \right) \hat{u}_p \right] \quad (5.37) \end{aligned}$$

and finally substituting in (5.35) gives the final result:

$$\begin{aligned} H_0(j\omega) = & \frac{1}{T_s} \left\{ \left[ \left( C_1^T \eta_1 + \{ [C_2^T \eta_2 + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \beta_2] \Gamma_1 - \zeta_1 \mu_1 h_1^T \} \beta_1 \right) \right. \right. \\ & \times (I - \Gamma_3 \beta_3 \beta_2 \Gamma_1 \beta_1)^{-1} \xi_3 \mu_3 + \zeta_3 \mu_3 \left. \right] \hat{r}_p + \left[ \left( C_1^T \eta_1 + \{ [C_2^T \eta_2 \right. \right. \\ & + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \beta_2] \Gamma_1 - \zeta_1 \mu_1 h_1^T \} \beta_1 \right) (I - \Gamma_3 \beta_3 \beta_2 \Gamma_1 \beta_1)^{-1} \\ & \times \Gamma_3 [\beta_3 (\beta_2 \Gamma_1 \eta_1 B_1 + \eta_2 B_2) + \eta_3 B_3] + \{ [C_2^T \eta_2 + (C_3^T \eta_3 \\ & - \zeta_3 \mu_3 h_3^T \beta_3) \beta_2] \Gamma_1 - \zeta_1 \mu_1 h_1^T \} \eta_1 B_1 + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \eta_2 B_2 \\ & \left. \left. - \zeta_3 \mu_3 h_3^T \eta_3 B_3 + \rho_1 + \rho_2 + \rho_3 \right] \hat{u}_p \right\} \quad (5.38) \end{aligned}$$

Rotating the subscripts in (5.38) to Model 1:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left\{ \left[ \left( C_2^T \eta_2 + \{ [C_3^T \eta_3 + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 - \zeta_2 \mu_2 h_2^T \} \beta_2 \right) \right. \right. \\
& \times (I - \Gamma_1 \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} \xi_1 \mu_1 + \zeta_1 \mu_1 \left. \right] \hat{r}_p + \left[ \left( C_2^T \eta_2 + \{ [C_3^T \eta_3 \right. \right. \\
& \times + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 - \zeta_2 \mu_2 h_2^T \} \beta_2 \right) (I - \Gamma_1 \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} \\
& \times \Gamma_1 [\beta_1 (\beta_1 \Gamma_2 \eta_2 B_2 + \eta_3 B_3) + \eta_1 B_1] + \{ [C_3^T \eta_3 + (C_1^T \eta_1 \\
& - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 - \zeta_2 \mu_2 h_2^T \} \eta_2 B_2 + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \eta_3 B_3 \\
& \left. \left. - \zeta_1 \mu_1 h_1^T \eta_1 B_1 + \rho_1 + \rho_2 + \rho_3 \right] \hat{u}_p \right\} \quad (5.39)
\end{aligned}$$

where

$$\begin{aligned}
\mu_1 &= \frac{1}{h_1^T \gamma_1 + m} \\
\mu_2 &= \frac{1}{h_2^T \gamma_2}
\end{aligned} \quad (5.40)$$

and  $h_1^T$  and  $h_2^T$  are vectors such that:

$$\begin{aligned}
h_1^T x(t) &= i_Q(t) \\
h_2^T x(t) &= i_P(t)
\end{aligned} \quad (5.41)$$

Setting  $\hat{r}_p = 1$  and  $\hat{u}_p = 0$  gives the control-to-output transfer function

$\frac{\hat{v}_{out}}{\hat{r}}$ :

$$\begin{aligned}
\frac{\hat{v}_{out}}{\hat{r}} = & \frac{\mu_1}{T_s} \left[ \left( C_2^T \eta_2 + \{ [C_3^T \eta_3 + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 \right. \right. \\
& \left. \left. - \zeta_2 \mu_2 h_2^T \} \beta_2 \right) (I - \Gamma_1 \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} \xi_1 + \zeta_1 \right] \quad (5.42)
\end{aligned}$$

By setting  $\hat{r} = 0$  and  $\hat{u}_p = 1$  the input-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{v}_g}$  can be found to be:

$$\begin{aligned} \frac{\hat{v}_{out}}{\hat{v}_g} = & \frac{1}{T_s} \left[ \left( C_2^T \eta_2 + \{ [C_3^T \eta_3 + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 - \zeta_2 \mu_2 h_2^T \} \beta_2 \right) \right. \\ & \times (I - \Gamma_1 \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} \Gamma_1 [\beta_1 (\beta_3 \Gamma_2 \eta_2 B_2 + \eta_3 B_3) + \eta_1 B_1] \\ & + \{ [C_3^T \eta_3 + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 - \zeta_2 \mu_2 h_2^T \} \eta_2 B_2 + (C_1^T \eta_1 \\ & \left. - \zeta_1 \mu_1 h_1^T \beta_1) \eta_3 B_3 - \zeta_1 \mu_1 h_1^T \eta_1 B_1 + \rho_1 + \rho_2 + \rho_3 \right] \quad (5.43) \end{aligned}$$

The control-to-output and input-to-output transfer functions are compared with results from simulation in Fig. 32 and Fig. 33, respectively, with the following parameters:

$$\begin{aligned} T_s &= 10\mu s \\ V_g &= 15V \\ D &= 0.25 \\ R_l &= 150\Omega \\ R_s &= 0.5\Omega \\ m &= 0V/s \end{aligned} \quad (5.44)$$

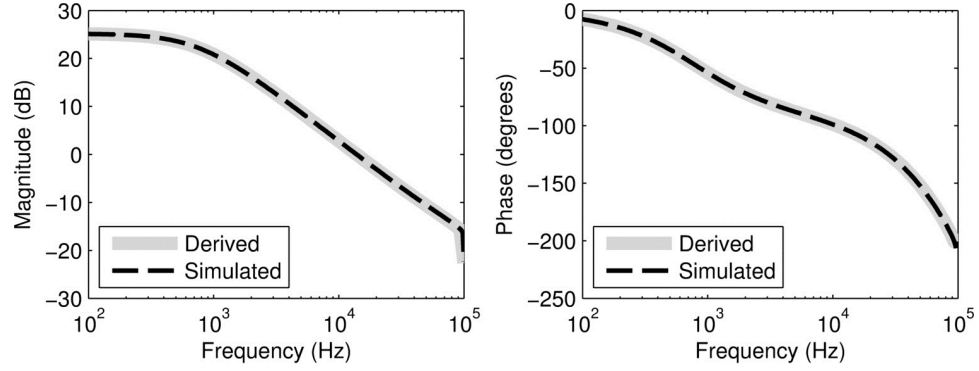


Figure 32: The derived current programmed boost converter in DCM control-to-output transfer function (solid) compared with results obtained from simulation (dashed).

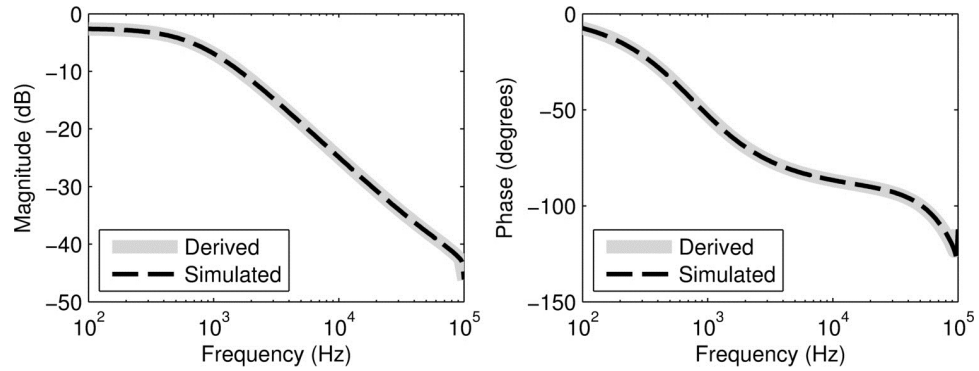


Figure 33: The derived current programmed boost converter in DCM input-to-output transfer function (solid) compared with results obtained from simulation (dashed).



### Section 5.3: Current Programmed Converter in DVM

Just like a NVM power converter, a CPM power converter operates in DVM when the voltage  $v_1(t)$  across the storage capacitor ( $C_1$  in the OTB converter) goes to zero, causing the switching diode  $P$  to turn on prematurely and creating a new topology before  $r(t)$  dictates the shutting off of  $Q$ . With  $N_s = 3$ ,  $T_s$  is split into three subintervals with DC lengths  $D_1T_s$ ,  $D_2T_s$ , and  $D_3T_s$  as shown in Fig. 34 which illustrates the switching period  $T_s$  for DVM described by both Model 1 and Model 2. The effective topologies of the OTB converter's power stage during the three subintervals are shown in Fig. 35. Model 2 will now be considered for the derivations. Notice that Model 2 starts two subintervals ahead of Model 1 in DVM, to ensure that  $r(t)$  affects both  $t_1$  and  $t_4$ .

$t_2$  is determined by the period  $T_s$  of the clock signal CLK and is unaffected by variations in  $r(t)$  and  $u(t)$ , so  $\hat{t}_2 = 0$ .  $t_3$  is the time when the voltage across  $C_1$ ,  $v_1(t)$ , goes to zero:

$$v_1(t_3) = 0 \quad (5.45)$$

Defining a vector  $h^T$  such that:

$$h_2^T x(t) = v_1(t) \quad (5.46)$$

allows (5.45) to be expressed as:

$$h_2^T x(t_3) = 0 \quad (5.47)$$

Dropping the DC component of  $x(t_3)$ :

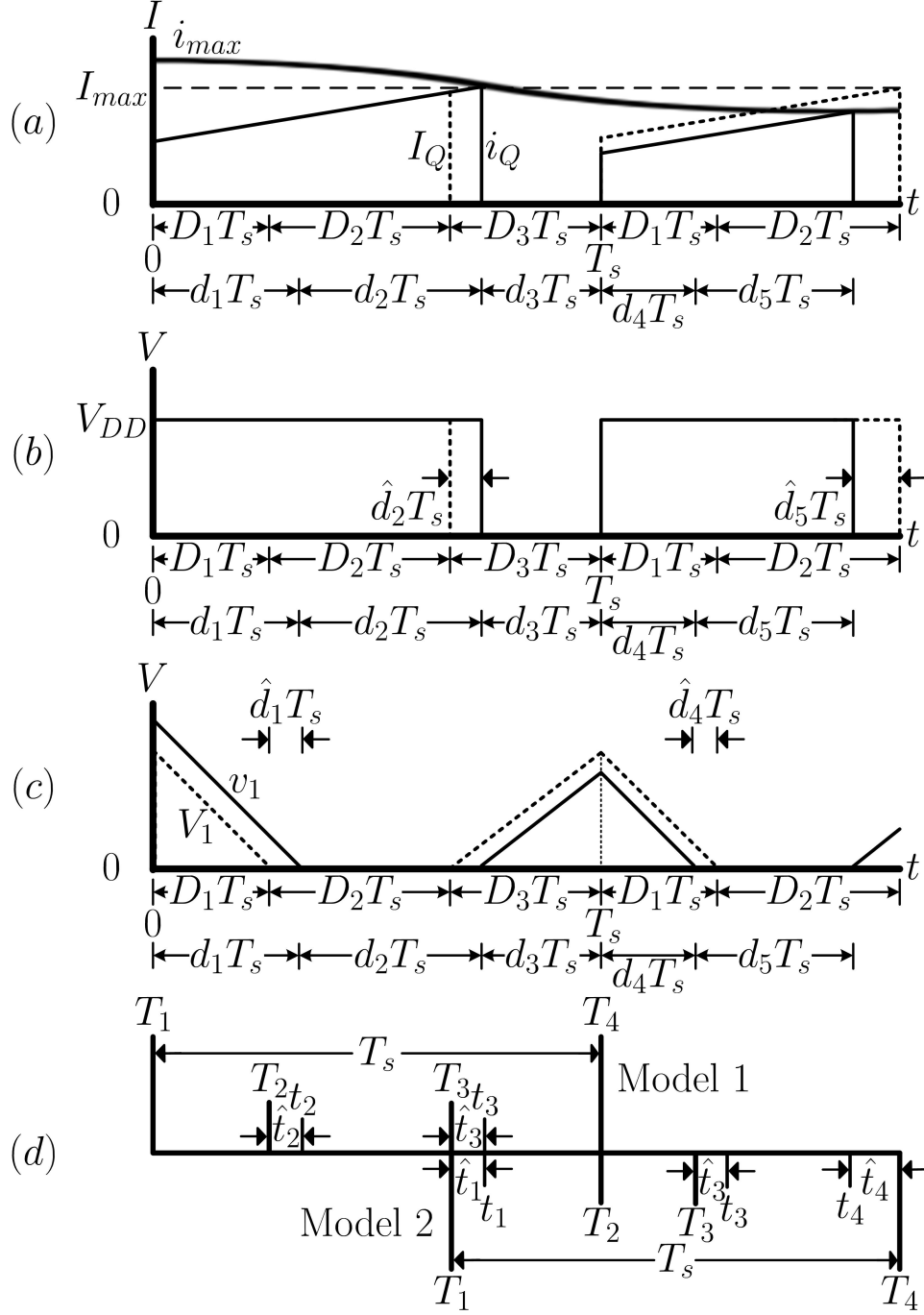


Figure 34: Switching period  $T_s$  for a CPM converter in DVM, described for both Model 1 and Model 2. (a) shows the current  $i_Q(t)$  being compared to a maximum current  $i_{max}(t)$  to produce the square wave (b). (c) shows the storage capacitor voltage  $v_1(t)$  reaching zero and creating a third intermediate topology. (d) illustrates how the times are denoted under Model 1 and Model 2.

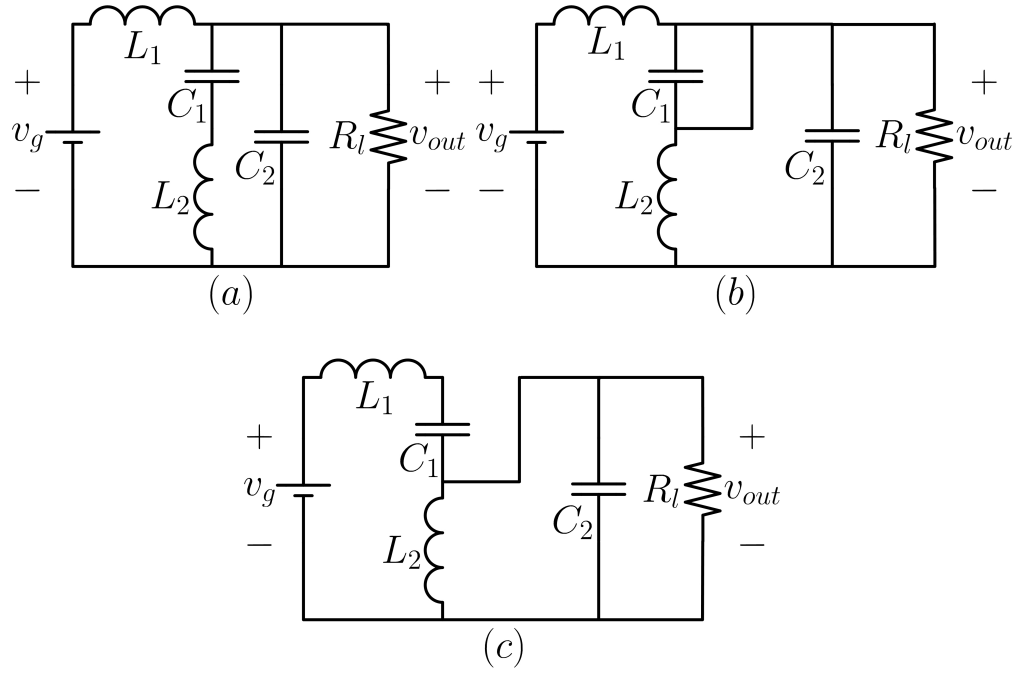


Figure 35: The effective topologies of a CPM OTB converter in DVM assuming ideal switches. (a) is the effective topology during the first subinterval of DC length  $D_1T_s$  where  $Q$  is on and  $P$  is off. (b) is the effective topology during the second subinterval of DC length  $D_2T_s$  where both  $Q$  and  $P$  are on. (c) is the effective topology during the third subinterval of DC length  $D_3T_s$  where  $Q$  is off and  $P$  is on.

$$h_2^T \hat{x}(t_3) = 0 \quad (5.48)$$

Substituting in (2.19) and solving for  $\hat{t}_3$ :

$$\hat{t}_3 = -\mu_2 h_2^T \Phi_2 \hat{x}(T_2) - \mu_2 h_2^T \eta_2 B_2 \hat{u}(T_3) \quad (5.49)$$

where

$$\mu_2 = \frac{1}{h_2^T \gamma_2} \quad (5.50)$$

The expression for  $\hat{t}_4$  is the same as the expression for the  $\hat{t}_4$  of the CPM converter in DCM case:

$$\hat{t}_4 = \mu_3 \hat{r}(T_4) - \mu_3 h_3^T \Phi_3 \hat{x}(T_3) - \mu_3 h_3^T \eta_3 B_3 \hat{u}(T_4) \quad (5.51)$$

where

$$\mu_3 = \frac{1}{h_3^T \gamma_3 + m} \quad (5.52)$$

$$h_3^T x(t) = i_Q(t) \quad (5.53)$$

Finding an expression for  $\hat{x}(T_1)$  is the next step after obtaining one for  $\hat{t}_4$ . By using (2.27),  $\hat{x}(T_2)$  is:

$$\hat{x}(T_2) = \Phi_1 \hat{x}(T_1) + \eta_1 B_1 \hat{u}(T_2) \quad (5.54)$$

Using (2.27) again and substituting in (5.49) and (5.54) gives the following expression for  $\hat{x}(T_3)$ :

$$\begin{aligned}
\hat{x}(T_3) &= \Phi_2 \hat{x}(T_2) + \xi_2 \hat{t}_3 + \eta_2 B_2 \hat{u}(T_3) \\
&= \Gamma_2 \Phi_2 \hat{x}(T_2) + \Gamma_2 \eta_2 B_2 \hat{u}(T_3) \\
&= \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}(T_2) + \Gamma_2 \eta_2 B_2 \hat{u}(T_3)
\end{aligned} \tag{5.55}$$

where

$$\Gamma_i = I - \xi_i \mu_i h_i^T \tag{5.56}$$

Using (2.27) once more to find  $\hat{x}(T_3)$  and substituting in (5.51) and (5.55):

$$\begin{aligned}
\hat{x}(T_4) &= \Phi_3 \hat{x}(T_3) + \xi_3 \hat{t}_4 + \eta_3 B_3 \hat{u}(T_4) \\
&= \Gamma_3 \Phi_3 \hat{x}(T_3) + \xi_3 \mu_3 \hat{r}(T_4) + \Gamma_3 \eta_3 B_3 \hat{u}(T_4) \\
&= \Gamma_3 \Phi_3 \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}(T_4) + \Gamma_3 \Phi_3 \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}(T_2) \\
&\quad + \Gamma_3 \Phi_3 \Gamma_2 \eta_2 B_2 \hat{u}(T_3) + \Gamma_3 \eta_3 B_3 \hat{u}(T_4)
\end{aligned} \tag{5.57}$$

Expressing all of the signals in their exponential form as in (2.31) and substituting in (2.32) for  $\hat{x}(T_4)$ :

$$\begin{aligned}
\hat{x}(T_1) e^{j\omega T_s} &= \Gamma_3 \Phi_3 \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}_p e^{j\omega T_4} \\
&\quad + \Gamma_3 \Phi_3 \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}_p e^{j\omega T_2} + \Gamma_3 \Phi_3 \Gamma_2 \eta_2 B_2 \hat{u}_p e^{j\omega T_3} \\
&\quad + \Gamma_3 \eta_3 B_3 \hat{u}_p e^{j\omega T_4}
\end{aligned} \tag{5.58}$$

Solving for  $\hat{x}(T_1)$ :

$$\begin{aligned}
\hat{x}(T_1) &= (I - \Gamma_3 \beta_3 \Gamma_2 \beta_2 \beta_1)^{-1} \{ \xi_3 \mu_3 \hat{r}(T_1) + [\Gamma_3 \beta_3 \Gamma_2 (\beta_2 \eta_1 B_1 \\
&\quad + \eta_2 B_2) + \Gamma_3 \eta_3 B_3] \hat{u}(T_1) \}
\end{aligned} \tag{5.59}$$

Now what is left is to substitute (5.51), (5.54), (5.55), and (5.59) into (3.23).

Expanding (3.23) for  $N_s = 3$ :

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} [C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \rho_1 \hat{u}_p + C_2^T \eta_2 \hat{x}(T_2) e^{-j\omega T_2} \\
& + \zeta_2 \hat{t}_3 e^{-j\omega T_3} + \rho_2 \hat{u}_p + C_3^T \eta_3 \hat{x}(T_3) e^{-j\omega T_3} + \zeta_3 \hat{t}_4 e^{-j\omega T_4} \\
& + \rho_3 \hat{u}_p] \quad (5.60)
\end{aligned}$$

Substituting in (5.49), (5.51), (5.54), and (5.55) gives:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left( C_1^T \eta_1 + \{C_2^T \eta_2 + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 - \zeta_2 \mu_2 h_2^T] \beta_2\} \beta_1 \right) \\
& \times \hat{x}(T_1) e^{j\omega T_1} + \zeta_3 \mu_3 \hat{r}_p + \left( \{C_2^T \eta_2 + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 \right. \\
& - \zeta_2 \mu_2 h_2^T] \beta_2\} \eta_1 B_1 + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 - \zeta_2 \mu_2 h_2^T] \eta_2 B_2 \\
& \left. - \zeta_3 \mu_3 h_3^T \eta_3 B_3 + \rho_1 + \rho_2 + \rho_3 \right) \hat{u}_p \quad (5.61)
\end{aligned}$$

Finally substituting in (5.59):

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left\{ \left[ \left( C_1^T \eta_1 + \{C_2^T \eta_2 + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 - \zeta_2 \mu_2 h_2^T] \beta_2\} \right. \right. \right. \\
& \times \beta_1 \Big) (I - \Gamma_3 \beta_3 \Gamma_2 \beta_2 \beta_1)^{-1} \zeta_3 \mu_3 + \zeta_3 \mu_3 \Big] \hat{r}_p + \left[ \left( C_1^T \eta_1 + \{C_2^T \eta_2 \right. \right. \\
& + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 - \zeta_2 \mu_2 h_2^T] \beta_2\} \beta_1 \Big) (I - \Gamma_3 \beta_3 \Gamma_2 \beta_2 \beta_1)^{-1} \\
& \times (\Gamma_3 \beta_3 \Gamma_2 \beta_2 \eta_1 B_1 + \Gamma_3 \beta_3 \Gamma_2 \eta_2 B_2 + \Gamma_3 \eta_3 B_3) + \{C_2^T \eta_2 + [(C_3^T \eta_3 \\
& - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 - \zeta_2 \mu_2 h_2^T] \beta_2\} \eta_1 B_1 + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 \\
& \left. - \zeta_2 \mu_2 h_2^T] \eta_2 B_2 - \zeta_3 \mu_3 h_3^T \eta_3 B_3 + \rho_1 + \rho_2 + \rho_3 \right] \hat{u}_p \Big\} \quad (5.62)
\end{aligned}$$

In DVM, rotating the subscripts back to Model 1 requires two rotations such

that 1 is replaced with 3, 2 with 1, and 3 with 2:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left\{ \left[ \left( C_3^T \eta_3 + \{ C_1^T \eta_1 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 - \zeta_1 \mu_1 h_1^T] \beta_1 \} \right. \right. \right. \\
& \times \beta_3 \Big) (I - \Gamma_2 \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} \xi_2 \mu_2 + \zeta_2 \mu_2 \Big] \hat{r}_p + \left[ \left( C_3^T \eta_3 + \{ C_1^T \eta_1 \right. \right. \\
& + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 - \zeta_1 \mu_1 h_1^T] \beta_1 \} \beta_3 \Big) (I - \Gamma_2 \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} \\
& \times (\Gamma_2 \beta_2 \Gamma_1 \beta_1 \eta_3 B_3 + \Gamma_2 \beta_2 \Gamma_1 \eta_1 B_1 + \Gamma_2 \eta_2 B_2) + \{ C_1^T \eta_1 + [(C_2^T \eta_2 \\
& - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 - \zeta_1 \mu_1 h_1^T] \beta_1 \} \eta_3 B_3 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 \\
& \left. \left. \left. - \zeta_1 \mu_1 h_1^T] \eta_1 B_1 - \zeta_2 \mu_2 h_2^T \eta_2 B_2 + \rho_1 + \rho_2 + \rho_3 \right] \hat{u}_p \right\} \quad (5.63)
\end{aligned}$$

where

$$\begin{aligned}
\mu_1 &= \frac{1}{h_1^T \gamma_1} \\
\mu_2 &= \frac{1}{h_2^T \gamma_2 + m}
\end{aligned} \quad (5.64)$$

and where  $h_1^T$  and  $h_2^T$  are vectors such that:

$$\begin{aligned}
h_1^T x(t) &= v_1(t) \\
h_2^T x(t) &= i_Q(t)
\end{aligned} \quad (5.65)$$

By setting  $\hat{r}_p = 1$  and  $\hat{u}_p = 0$  the control-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{r}}$  can be found:

$$\begin{aligned}
\frac{\hat{v}_{out}}{\hat{r}} = & \frac{\mu_2}{T_s} \left[ \left( C_3^T \eta_3 + \{ C_1^T \eta_1 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 \right. \right. \\
& \left. \left. - \zeta_1 \mu_1 h_1^T] \beta_1 \} \beta_3 \right) (I - \Gamma_2 \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} \xi_2 + \zeta_2 \right] \quad (5.66)
\end{aligned}$$

Setting  $\hat{r}_p = 0$  and  $\hat{u}_p = 1$ , the input-to-output transfer function  $\frac{v_{out}}{\hat{v}_g}$  is:

$$\begin{aligned} \frac{\hat{v}_{out}}{\hat{v}_g} = \frac{1}{T_s} & \left[ \left( C_3^T \eta_3 + \{ C_1^T \eta_1 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 \right. \right. \\ & \quad \left. \left. - \zeta_1 \mu_1 h_1^T] \beta_1 \} \beta_3 \right) (I - \Gamma_2 \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} (\Gamma_2 \beta_2 \Gamma_1 \beta_1 \eta_3 B_3 \right. \\ & \quad \left. + \Gamma_2 \beta_2 \Gamma_1 \eta_1 B_1 + \Gamma_2 \eta_2 B_2) + \{ C_1^T \eta_1 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 \right. \\ & \quad \left. - \zeta_1 \mu_1 h_1^T] \beta_1 \} \eta_3 B_3 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 - \zeta_1 \mu_1 h_1^T] \eta_1 B_1 \right. \\ & \quad \left. - \zeta_2 \mu_2 h_2^T \eta_2 B_2 + \rho_1 + \rho_2 + \rho_3 \right] \quad (5.67) \end{aligned}$$

The control-to-output and input-to-output transfer functions are compared with results from simulation in Fig. 36 and Fig. 37, respectively, with the following parameters:

$$\begin{aligned} T_s &= 10\mu s \\ V_g &= 28V \\ D &= 0.4 \\ R_l &= 3\Omega \\ R_s &= 0.5\Omega \\ m &= 0V/s \end{aligned} \quad (5.68)$$



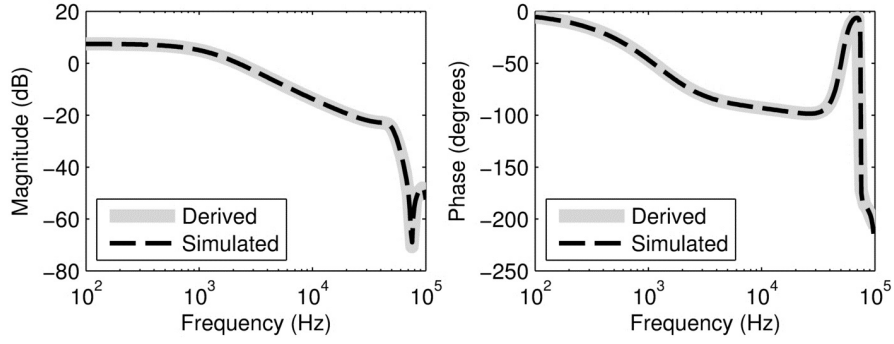


Figure 36: The derived CPM OTB converter in DVM control-to-output transfer function (solid) compared with results obtained from simulation (dashed).

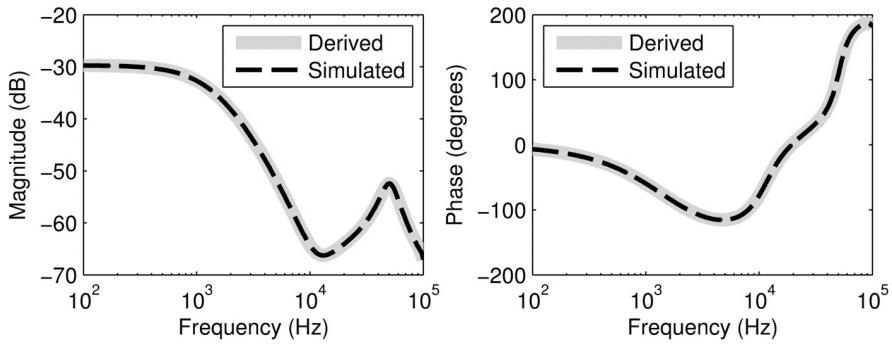


Figure 37: The derived CPM OTB converter in DVM input-to-output transfer function (solid) compared with results obtained from simulation (dashed).

## Chapter 6: $V^2$ Controlled Transfer Functions

A  $V^2$  controlled power converter senses the output voltage  $v_{out}(t)$  through a gain  $H(s)$  and uses it as the sawtooth signal as well as feeds it to the output voltage controller to generate  $r(t)$ . Using the output voltage to generate both the sawtooth and control signal allows for a very fast load transient response. Similar to a CPM converter,  $r(t)$  represents the peak output voltage  $v_{max}(t)$  the DC value of which,  $V_{max}$ , is related to  $V_{ref}$  by:

$$V_{ref} = V_{max}H(s) \quad (6.1)$$

Also similar to a CPM converter, an SR-latch is used to prevent the topologies from switching prematurely by having a clock signal CLK, which has an impulse every  $T_s$  seconds, fed into the set input S and the output of the comparator fed into the reset input R that turns  $Q$  off when  $v_{out}(t) = v_{max}(t)$ . The transfer functions of interest will be derived for the  $V^2$  controlled boost converter in Fig. 38 under CCM and DCM, followed by the  $V^2$  controlled OTB converter in Fig. 39 under DVM. The output voltage controllers have been omitted from Fig. 38 and Fig. 39 since they have no effect on the transfer functions of interest, however they are present and generate  $r(t)$ . No converters under  $V^2$  control have had their transfer functions of interest published to this accuracy.

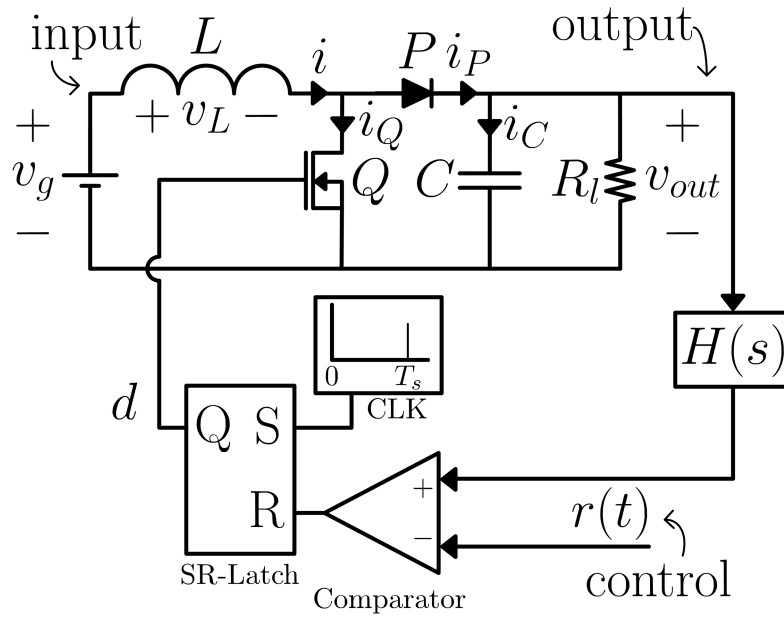


Figure 38:  $V^2$  controlled boost converter schematic. The output voltage  $v_{out}(t)$  is compared to a peak voltage  $v_{max}(t)$  which is contained in the signal  $r(t)$ . The output of the comparator, the clock signal  $CLK$ , and the SR-Latch produce a square wave with a period  $T_s$  and duty ratio  $d$ .

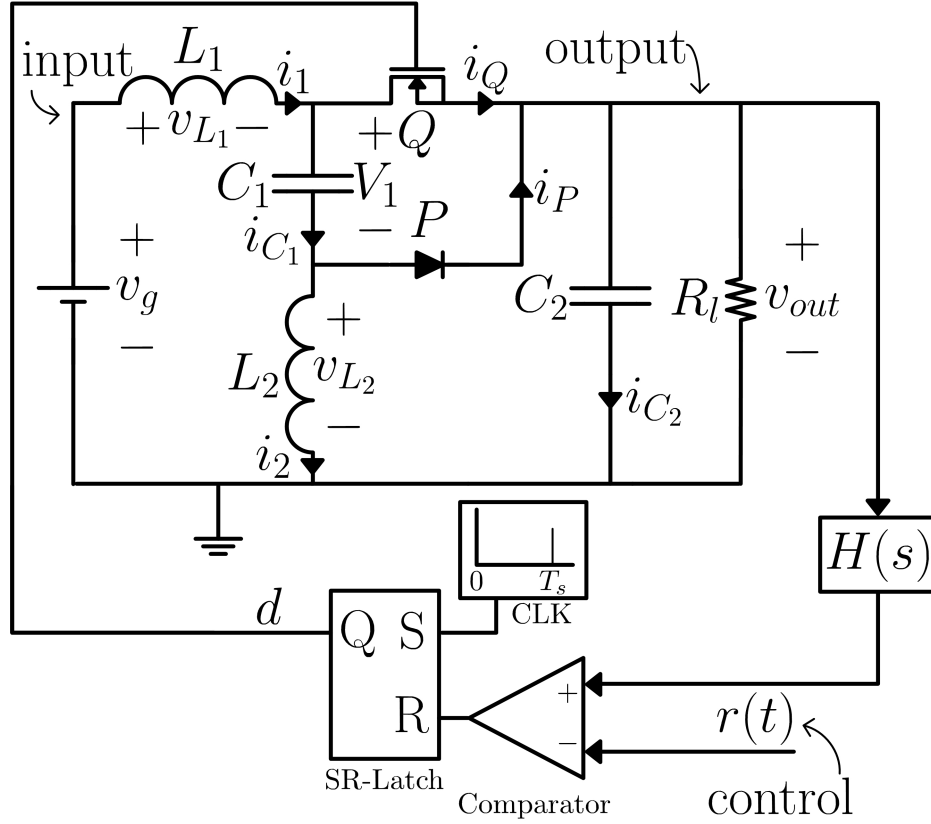


Figure 39:  $V^2$  controlled OTB converter schematic. The output voltage  $v_{out}(t)$  is compared to a peak voltage  $v_{max}(t)$  which is contained in the signal  $r(t)$ . The output of the comparator, the clock signal  $CLK$ , and the SR-Latch produce a square wave with a period  $T_s$  and duty ratio  $d$ .

## Section 6.1: $V^2$ Controlled Converter in CCM

A  $V^2$  controlled converter in CCM has only two topologies ( $N_s = 2$ ) and therefore has two subintervals over the period  $T_s$ . The first subinterval will have a DC length  $DT_s$  and the second subinterval will have a DC length  $(1 - D)T_s$ . The effective topologies of the boost power stage during the two subintervals are shown in Fig. 40. Under Model 1,  $DT_s$  is equal to the time it takes for  $V_{out}(t)$  to reach  $V_{max}$  where  $V_{out}(t)$  and  $V_{max}$  are the DC components of  $v_{out}(t)$  and  $v_{max}(t)$ , respectively. As with CPM, a stability ramp of slope  $m$  is added to the sawtooth signal so that the converter remains stable for  $D > 0.5$ , the effect of which is illustrated in Fig. 41. Fig. 42 shows the period  $T_s$  for both Model 1 and Model 2, the latter of which will now be considered for the following derivations until the subscripts of the final result are swapped back to become applicable to Model 1.

$t_2$  is determined by the CLK signal and is not sensitive to variations in  $r(t)$  or  $u(t)$ , therefore  $\hat{t}_2 = 0$ . Finding an expression for  $\hat{t}_3$  is accomplished by examining the switching condition:

$$r(t_3) = h^T x(t_3) + mt_3 \quad (6.2)$$

where  $h^T$  is a vector such that:

$$v_{out}(t)H(s) = h^T x(t) \quad (6.3)$$

Dropping all DC terms of (6.2) gives:

$$\hat{r}(T_3) = h^T \hat{x}(t_3) + m\hat{t}_3 \quad (6.4)$$

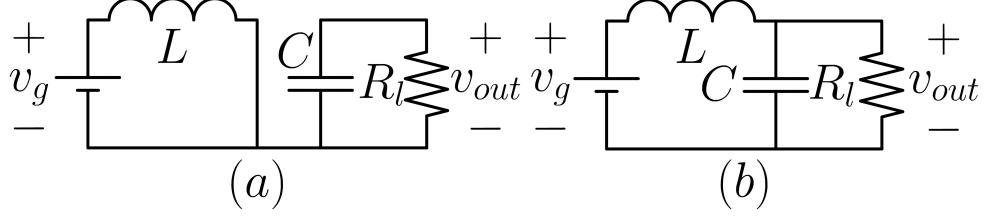


Figure 40: The effective topologies of a  $V^2$  controlled boost converter in CCM assuming ideal switches. (a) is the effective topology during the first subinterval of DC length  $DT_s$  where  $Q$  is on and  $P$  is off. (b) is the effective topology during the second subinterval of DC length  $(1 - D)T_s$  where  $Q$  is off and  $P$  is on.

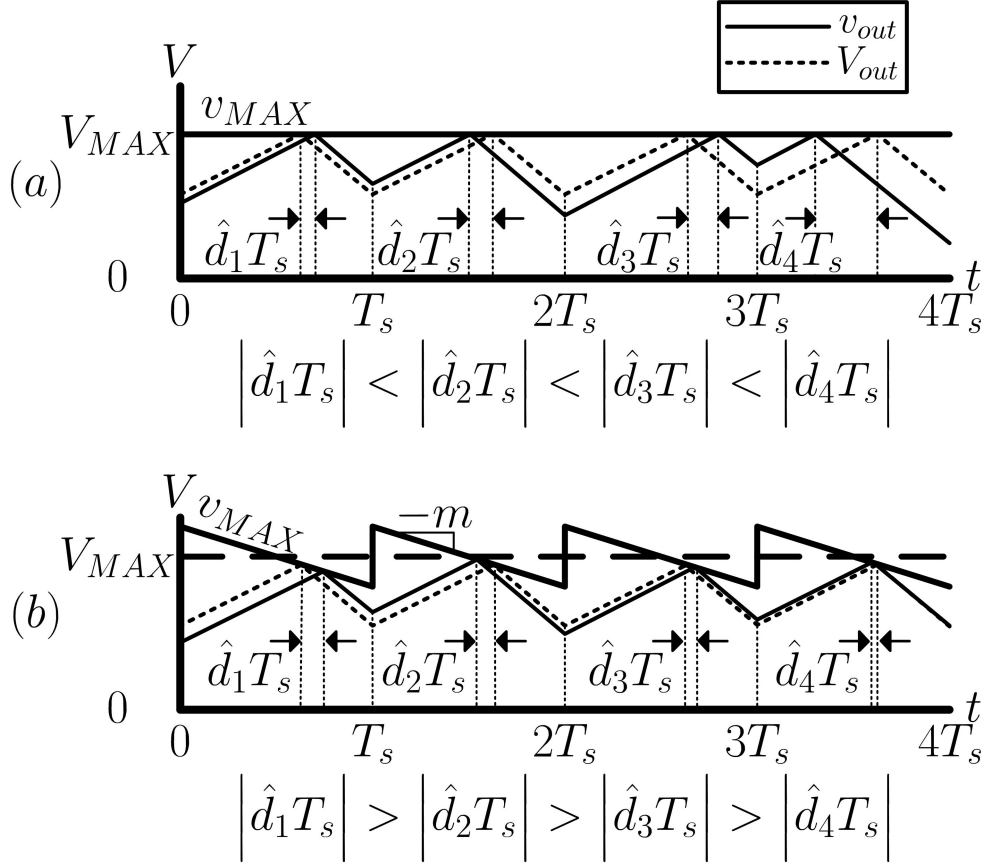


Figure 41: (a) shows a the potential instability for a  $V^2$  controlled converter with  $D > 0.5$  by showing the increase in the magnitudes of  $d_i$  over multiple switching periods  $T_s$ . (b) shows that an added stability ramp of slope  $m$  makes the voltage  $v_{out}(t)$  settle back to its average (DC)  $V_{out}(t)$  as time progresses, such that  $d_i$  decreases for increasing  $i$ .

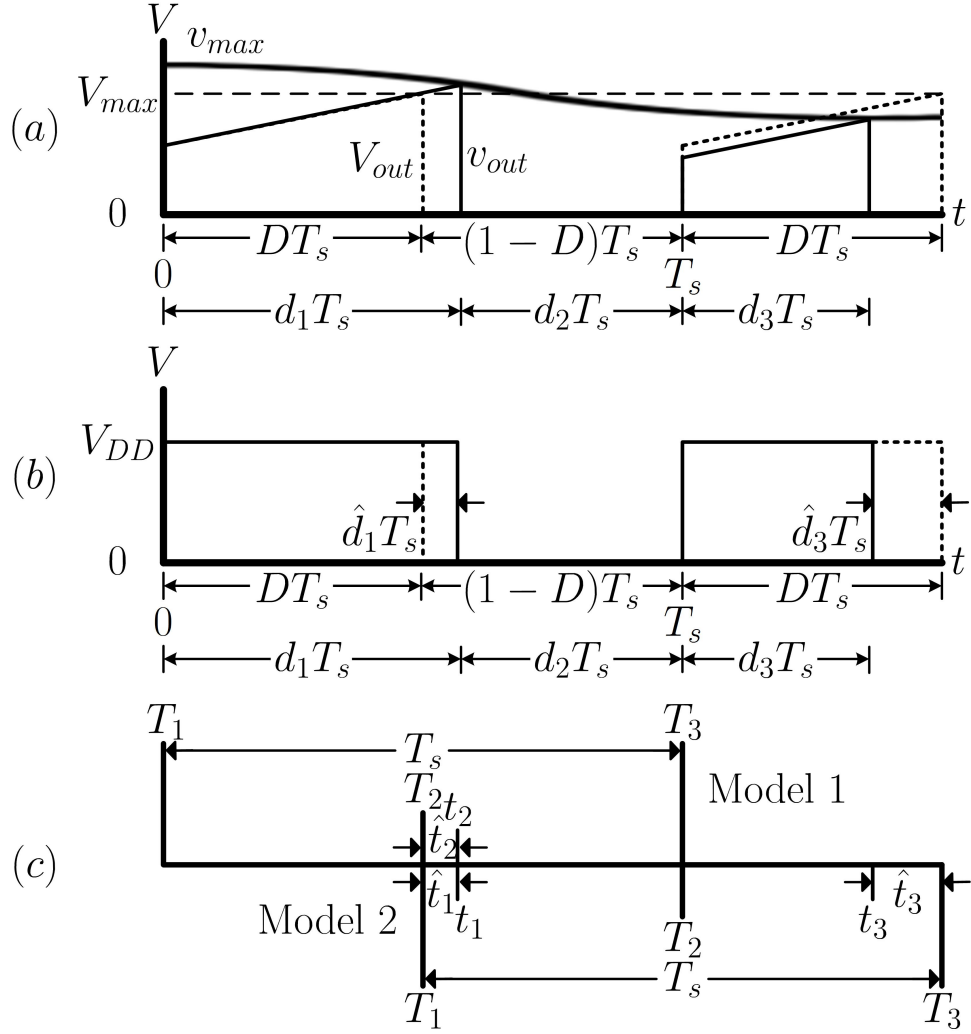


Figure 42: Switching period  $T_s$  for a  $V^2$  controlled converter in CCM, described for both Model 1 and Model 2. (a) shows the output voltage  $v_{out}(t)$  being compared to the peak output voltage  $v_{max}(t)$  to produce the square wave (b). (c) illustrates how the times are denoted under Model 1 and Model 2.

Substituting in (2.19) for  $\hat{x}(t_3)$  and solving for  $\hat{t}_3$  results in:

$$\hat{t}_3 = \mu \hat{r}(T_3) - \mu h^T \Phi_2 \hat{x}(T_2) - \mu h^T \eta_2 B_2 \hat{u}(T_3) \quad (6.5)$$

where

$$\mu = \frac{1}{h^T \gamma_2 + m} \quad (6.6)$$

With an expression for  $\hat{t}_3$  the next step is to get an expression for  $\hat{x}(T_1)$ . This is done by solving the small signal difference equation (2.27) by going through one switching period  $T_s$ .  $\hat{x}(T_2)$  is given by:

$$\hat{x}(T_2) = \Phi_1 \hat{x}(T_1) + \eta_1 B_1 \hat{u}(T_2) \quad (6.7)$$

Using (2.27) once again to find  $\hat{x}(T_3)$  and substituting in (6.7) and (6.5) gives:

$$\begin{aligned} \hat{x}(T_3) &= \Phi_2 \hat{x}(T_2) + \xi_2 \hat{t}_3 + \eta_2 B_2 \hat{u}(T_3) \\ &= \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \xi_2 \mu \hat{r}(T_3) + \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}(T_2) + \Gamma_2 \eta_2 B_2 \hat{u}(T_3) \end{aligned} \quad (6.8)$$

where

$$\Gamma_2 = I - \xi_2 \mu h^T \quad (6.9)$$

Expressing all of the signals in their exponential form as in (2.31) and substituting in (2.32) for  $\hat{x}(T_3)$ :

$$\hat{x}(T_1) e^{j\omega T_s} = \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \xi_2 \mu \hat{r}_p e^{j\omega T_4} + \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}_p e^{j\omega T_2} + \Gamma_2 \eta_2 B_2 \hat{u}_p e^{j\omega T_4} \quad (6.10)$$

Solving for  $\hat{x}(T_1)$ :

$$\hat{x}(T_1) = (I - \Gamma_2 \beta_2 \beta_1)^{-1} [\xi_2 \mu \hat{r}(T_1) + \Gamma_2 (\beta_2 \eta_1 B_1 + \eta_2 B_2) \hat{u}(T_1)] \quad (6.11)$$



The last step is to substitute (6.5), (6.7) and (6.11) into (3.23). Expanding (3.23) results in:

$$H_0(j\omega) = \frac{1}{T_s} [C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \rho_1 \hat{u}_p + C_2^T \eta_2 \hat{x}(T_2) e^{-j\omega T_2} + \zeta_2 \hat{t}_3 e^{-j\omega T_3} + \rho_2 \hat{u}_p] \quad (6.12)$$

Substituting in (6.5):

$$H_0(j\omega) = \frac{1}{T_s} [C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \zeta_2 \mu \hat{r}_p + (C_2^T \eta_2 - \zeta_2 \mu h^T \beta_2) \hat{x}(T_2) e^{-j\omega T_2} + (\rho_1 + \rho_2 - \zeta_2 \mu h^T \eta_2 B_2) \hat{u}_p] \quad (6.13)$$

Now substituting in (6.7):

$$H_0(j\omega) = \frac{1}{T_s} \{ [C_1^T \eta_1 + (C_2^T \eta_2 - \zeta_2 \mu h^T \beta_2) \beta_1] \hat{x}(T_1) e^{-j\omega T_1} + \zeta_2 \mu \hat{r}_p + [(C_2^T \eta_2 - \zeta_2 \mu h^T \beta_2) \eta_1 B_1 - \zeta_2 \mu h^T \eta_2 B_2 + \rho_1 + \rho_2] \hat{u}_p \} \quad (6.14)$$

Finally substituting in (6.11):

$$H_0(j\omega) = \frac{1}{T_s} \left( \{ [C_1^T \eta_1 + (C_2^T \eta_2 - \zeta_2 \mu h^T \beta_2) \beta_1] (I - \Gamma_2 \beta_2 \beta_1)^{-1} \zeta_2 \mu + \zeta_2 \mu \} \hat{r}_p + \{ [C_1^T \eta_1 + (C_2^T \eta_2 - \zeta_2 \mu h^T \beta_2) \beta_1] (I - \Gamma_2 \beta_2 \beta_1)^{-1} \times [\Gamma_2 (\beta_2 \eta_1 B_1 + \eta_2 B_2)] + (C_2^T \eta_2 - \zeta_2 \mu h^T \beta_2) \eta_1 B_1 - \zeta_2 \mu h^T \eta_2 B_2 + \rho_1 + \rho_2 \} \hat{u}_p \right) \quad (6.15)$$

Rotating the subscripts so that the final result is in Model 1:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left( \{[C_2^T \eta_2 + (C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \beta_2](I - \Gamma_1 \beta_1 \beta_2)^{-1} \xi_1 \mu \right. \\
& + \zeta_1 \mu \} \hat{r}_p + \{[C_2^T \eta_2 + (C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \beta_2](I - \Gamma_1 \beta_1 \beta_2)^{-1} \\
& \times [\Gamma_1(\beta_1 \eta_2 B_2 + \eta_1 B_1)] + (C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \eta_2 B_2 \\
& \left. - \zeta_1 \mu h^T \eta_1 B_1 + \rho_1 + \rho_2 \} \hat{u}_p \right) \quad (6.16)
\end{aligned}$$

where

$$\mu = \frac{1}{h^T \gamma_1 + m} \quad (6.17)$$

As with the previous control methods, the transfer functions for  $V^2$  control are found by setting the perturbation of the input of interest to one while setting the other to zero. The control-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{r}}$  is found by setting  $\hat{r}_p = 1$  and setting the  $\hat{u}_p = 0$ :

$$\frac{\hat{v}_{out}}{\hat{r}} = \frac{\mu}{T_s} \{[(C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \beta_2 + C_2^T \eta_2](I - \Gamma_1 \beta_1 \beta_2)^{-1} \xi_1 + \zeta_1 \} \quad (6.18)$$

The input-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{v}_g}$  is found by setting  $\hat{r}_p = 0$  and  $\hat{u}_p = 1$ :

$$\begin{aligned}
\frac{\hat{v}_{out}}{\hat{v}_g} = & \frac{1}{T_s} \{[(C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \beta_2 + C_2^T \eta_2](I - \Gamma_1 \beta_1 \beta_2)^{-1} [\Gamma_1(\eta_1 B_1 \\
& + \beta_1 \eta_2 B_2)] - \zeta_1 \mu h^T \eta_1 B_1 + (C_1^T \eta_1 - \zeta_1 \mu h^T \beta_1) \eta_2 B_2 + \rho_1 + \rho_2 \} \quad (6.19)
\end{aligned}$$

The control-to-output and input-to-output transfer functions are compared with results from simulation in Fig. 43 and Fig. 44, respectively, with the following parameters:

$$\begin{aligned}
T_s &= 10\mu s \\
V_g &= 15V \\
D &= 0.25 \\
R_l &= 18.6\Omega \\
m &= 0V/s
\end{aligned}
\tag{6.20}$$

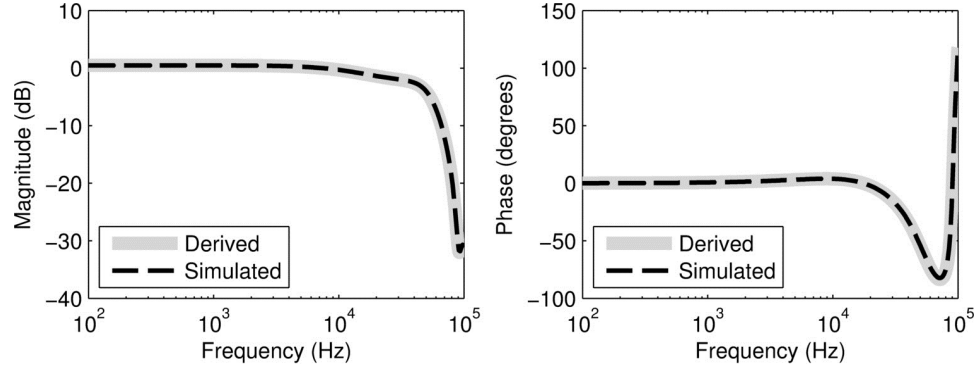


Figure 43: The derived  $V^2$  controlled boost converter in CCM control-to-output transfer function (solid) compared with results obtained from simulation (dashed).

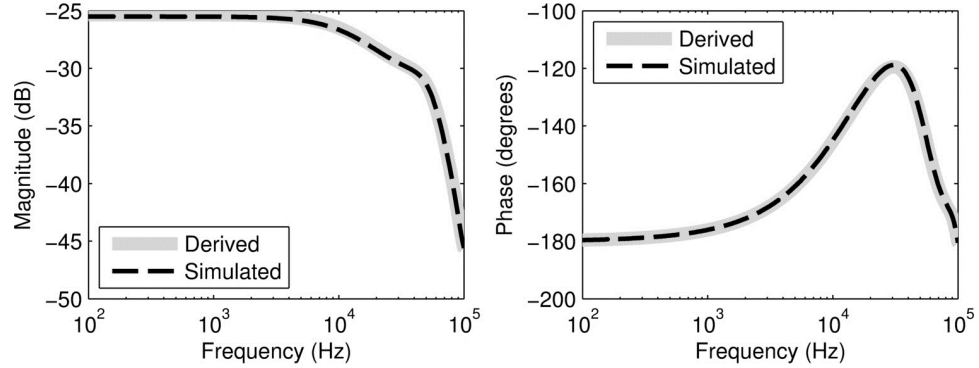


Figure 44: The derived  $V^2$  controlled boost converter in CCM input-to-output transfer function (solid) compared with results obtained from simulation (dashed).

## Section 6.2: $V^2$ Controlled Converter in DCM

As with the previously discussed control methods, a  $V^2$  controlled power converter operates in DCM when  $i_P(t)$  goes to zero and the switching diode  $P$  shuts off before  $Q$  is turned back on. This results in a third topology such that  $N_s = 3$ , so  $T_s$  is split into three subintervals of DC lengths  $D_1T_s$ ,  $D_2T_s$ , and  $D_3T_s$  shown by Fig. 45. The effective topologies of the boost power stage during the three subintervals are shown in Fig. 46. Model 2, as shown in Fig. 45, will be used for the following derivation, and then the subscripts of the final result will be rotated appropriately back to Model 1.

$t_2$  is when the switching diode current  $i_P(t)$  goes to zero:

$$i_P(t_2) = 0 \quad (6.21)$$

Defining a vector  $h_1^T$  such that:

$$h_1^T x(t) = i_P(t) \quad (6.22)$$

allows (6.21) to be expressed as:

$$h_1^T x(t_2) = 0 \quad (6.23)$$

Dropping the DC component of (6.23) gets:

$$h_1^T \hat{x}(t_2) = 0 \quad (6.24)$$

Substituting in (2.19) and solving for  $\hat{t}_2$ :

$$\hat{t}_2 = -\mu_1 h_1^T \Phi_1 \hat{x}(T_1) - \mu_1 h_1^T \eta_1 B_1 \hat{u}(T_2) \quad (6.25)$$

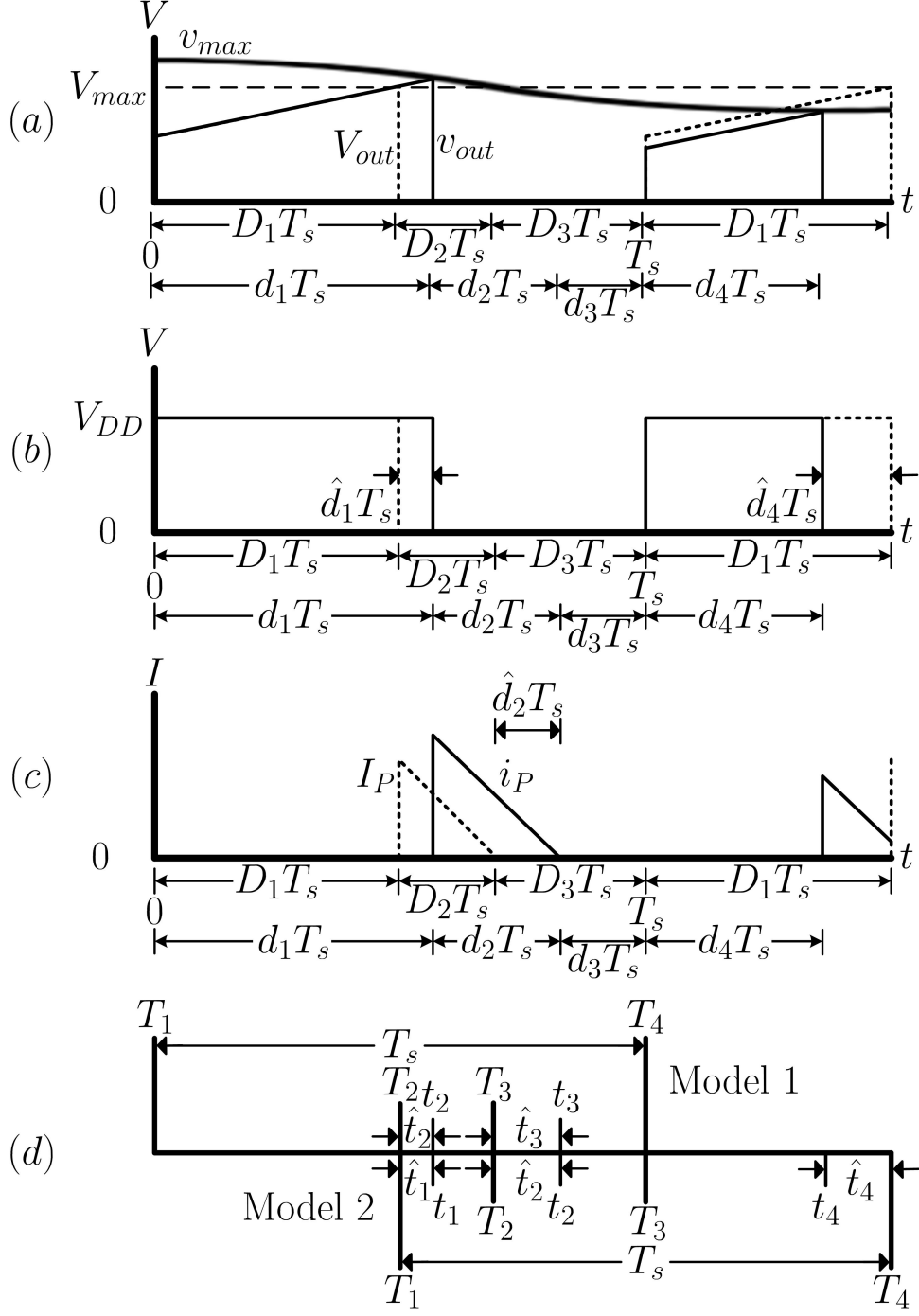


Figure 45: Switching period  $T_s$  for a  $V^2$  controlled converter in DCM, described for both Model 1 and Model 2. (a) shows the output voltage  $v_{out}(t)$  being compared to the peak output voltage  $v_{max}(t)$  to produce the square wave (b). (c) shows the diode current  $i_P(t)$  reaching zero and creating a third topology. (d) illustrates how the times are denoted under Model 1 and Model 2.

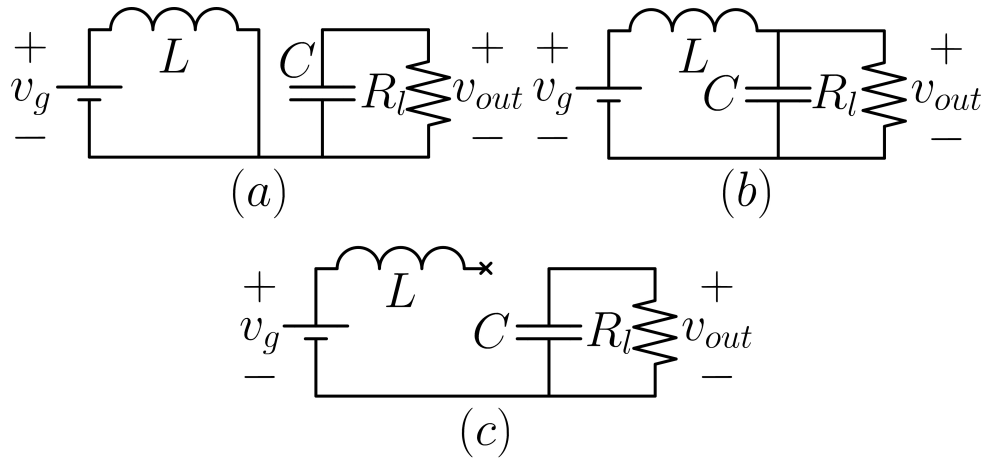


Figure 46: The effective topologies of a  $V^2$  controlled boost converter in DCM assuming ideal switches. (a) is the effective topology during the first subinterval of DC length  $D_1T_s$  where  $Q$  is on and  $P$  is off. (b) is the effective topology during the second subinterval of DC length  $D_2T_s$  where  $Q$  is off and  $P$  is on. (c) is the effective topology during the third subinterval of DC length  $D_3T_s$  where both  $Q$  and  $P$  are off.

where

$$\mu_1 = \frac{1}{h_1^T \gamma_1} \quad (6.26)$$

$t_3$  is determined by the period  $T_s$  of the clock signal CLK and is not sensitive to variations in  $r(t)$  or  $u(t)$ , therefore  $\hat{t}_3 = 0$ . Finding  $\hat{t}_4$  involves the same procedure used for  $\hat{t}_3$  in the  $V^2$  controlled converter in CCM case:

$$\hat{t}_4 = \mu_3 \hat{r}(T_4) - \mu_3 h_3^T \Phi_3 \hat{x}(T_3) - \mu_3 h_3^T \eta_3 B_3 \hat{u}(T_4) \quad (6.27)$$

where

$$\mu_3 = \frac{1}{h_3^T \gamma_3 + m} \quad (6.28)$$

$$h_3^T x(t) = v_{out}(t) H(s) \quad (6.29)$$

and  $m$  is the slope of the added stability ramp.

$\hat{x}(T_1)$  will be found next. By using (2.27) and substituting in (6.25), an expression for  $\hat{x}(T_2)$  is found:

$$\begin{aligned} \hat{x}(T_2) &= \Phi_1 \hat{x}(T_1) + \xi_1 \hat{t}_2 + \eta_1 B_1 \hat{u}(T_2) \\ &= \Gamma \Phi_1 \hat{x}(T_1) + \Gamma_1 \eta_1 B_1 \hat{u}(T_2) \end{aligned} \quad (6.30)$$

where

$$\Gamma_i = I - \xi_i \mu_i h_i^T \quad (6.31)$$

Using (2.27) again and substituting in (6.30) gets an expression for  $\hat{x}(T_3)$ :

$$\begin{aligned} \hat{x}(T_3) &= \Phi_2 \hat{x}(T_2) + \eta_2 B_2 \hat{u}(T_3) \\ &= \Phi_2 \Gamma_1 \Phi_1 \hat{x}(T_1) + \Phi_2 \Gamma_1 \eta_1 B_1 \hat{u}(T_2) + \eta_2 B_2 \hat{u}(T_3) \end{aligned} \quad (6.32)$$



Using (2.27) once more and substituting in (6.27) and (6.32) gives an expression for  $\hat{x}(T_4)$ :

$$\begin{aligned}
\hat{x}(T_4) &= \Phi_3 \hat{x}(T_3) + \eta_3 B_3 \hat{u}(T_4) \\
&= \Gamma_3 \Phi_3 \hat{x}(T_3) + \xi_3 \mu_3 \hat{r}(T_4) + \Gamma_3 \eta_3 B_3 \hat{u}(T_4) \\
&= \Gamma_3 \Phi_3 \Phi_2 \Gamma_1 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}(T_4) + \Gamma_3 \Phi_3 \Phi_2 \Gamma_1 \eta_1 B_1 \hat{u}(T_2) \\
&\quad + \Gamma_3 \Phi_3 \eta_2 B_2 \hat{u}(T_3) + \Gamma_3 \eta_3 B_3 \hat{u}(T_4)
\end{aligned} \tag{6.33}$$

Expressing all of the signals in their exponential form as in (2.31) and substituting in (2.32) for  $\hat{x}(T_4)$ :

$$\begin{aligned}
\hat{x}(T_1) e^{j\omega T_s} &= \Gamma_3 \Phi_3 \Phi_2 \Gamma_1 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}_p e^{j\omega T_4} + \Gamma_3 \Phi_3 \Phi_2 \Gamma_1 \eta_1 B_1 \hat{u}_p e^{j\omega T_2} \\
&\quad + \Gamma_3 \Phi_3 \eta_2 B_2 \hat{u}_p e^{j\omega T_3} + \Gamma_3 \eta_3 B_3 \hat{u}_p e^{j\omega T_4}
\end{aligned} \tag{6.34}$$

Solving for  $\hat{x}(T_1)$ :

$$\begin{aligned}
\hat{x}(T_1) &= (I - \Gamma_3 \beta_3 \beta_2 \Gamma_1 \beta_1)^{-1} \{ \xi_3 \mu_3 \hat{r}(T_1) + \Gamma_3 [ \beta_3 (\beta_2 \Gamma_1 \eta_1 B_1 \\
&\quad + \eta_2 B_2) + \eta_3 B_3 ] \hat{u}(T_1) \}
\end{aligned} \tag{6.35}$$

The final step is to substitute (6.27), (6.30), (6.32), and (6.35) into (3.23). Expanding (3.23) for  $N_s = 3$ :

$$\begin{aligned}
H_0(j\omega) &= \frac{1}{T_s} [ C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \zeta_1 \hat{t}_2 e^{-j\omega T_2} + \rho_1 \hat{u}_p + C_2^T \eta_2 \hat{x}(T_2) e^{-j\omega T_2} \\
&\quad + \rho_2 \hat{u}_p + C_3^T \eta_3 \hat{x}(T_3) e^{-j\omega T_3} + \zeta_3 \hat{t}_4 e^{-j\omega T_4} + \rho_3 \hat{u}_p ]
\end{aligned} \tag{6.36}$$

Substituting in (6.25), (6.27), (6.30), and (6.32):

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left[ \left( C_1^T \eta_1 + \{ [C_2^T \eta_2 + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \beta_2] \Gamma_1 - \zeta_1 \mu_1 h_1^T \} \beta_1 \right) \right. \\
& \times \hat{x}(T_1) e^{-j\omega T_1} + \zeta_3 \mu_3 \hat{r}_p + \left( \{ [C_2^T \eta_2 + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \beta_2] \Gamma_1 \right. \\
& - \zeta_1 \mu_1 h_1^T \} \eta_1 B_1 + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \eta_2 B_2 - \zeta_3 \mu_3 h_3^T \eta_3 B_3 \\
& \left. \left. + \rho_1 + \rho_2 + \rho_3 \right) \hat{u}_p \right] \quad (6.37)
\end{aligned}$$

and finally substituting in (6.35) gives the final result:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left\{ \left[ \left( C_1^T \eta_1 + \{ [C_2^T \eta_2 + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \beta_2] \Gamma_1 - \zeta_1 \mu_1 h_1^T \} \beta_1 \right) \right. \right. \\
& \times (I - \Gamma_3 \beta_3 \beta_2 \Gamma_1 \beta_1)^{-1} \xi_3 \mu_3 + \zeta_3 \mu_3 \left. \right] \hat{r}_p + \left[ \left( C_1^T \eta_1 + \{ [C_2^T \eta_2 \right. \right. \\
& + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \beta_2] \Gamma_1 - \zeta_1 \mu_1 h_1^T \} \beta_1 \right) (I - \Gamma_3 \beta_3 \beta_2 \Gamma_1 \beta_1)^{-1} \\
& \times \Gamma_3 [\beta_3 (\beta_2 \Gamma_1 \eta_1 B_1 + \eta_2 B_2) + \eta_3 B_3] + \{ [C_2^T \eta_2 + (C_3^T \eta_3 \\
& - \zeta_3 \mu_3 h_3^T \beta_3) \beta_2] \Gamma_1 - \zeta_1 \mu_1 h_1^T \} \eta_1 B_1 + (C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \eta_2 B_2 \\
& \left. \left. - \zeta_3 \mu_3 h_3^T \eta_3 B_3 + \rho_1 + \rho_2 + \rho_3 \right] \hat{u}_p \right\} \quad (6.38)
\end{aligned}$$

Rotating the subscripts in (6.38) to Model 1:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left\{ \left[ \left( C_2^T \eta_2 + \{ [C_3^T \eta_3 + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 - \zeta_2 \mu_2 h_2^T \} \beta_2 \right) \right. \right. \\
& \times (I - \Gamma_1 \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} \xi_1 \mu_1 + \zeta_1 \mu_1 \left. \right] \hat{r}_p + \left[ \left( C_2^T \eta_2 + \{ [C_3^T \eta_3 \right. \right. \\
& \times + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 - \zeta_2 \mu_2 h_2^T \} \beta_2 \right) (I - \Gamma_1 \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} \\
& \times \Gamma_1 [\beta_1 (\beta_3 \Gamma_2 \eta_2 B_2 + \eta_3 B_3) + \eta_1 B_1] + \{ [C_3^T \eta_3 + (C_1^T \eta_1 \\
& - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 - \zeta_2 \mu_2 h_2^T \} \eta_2 B_2 + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \eta_3 B_3 \\
& \left. \left. - \zeta_1 \mu_1 h_1^T \eta_1 B_1 + \rho_1 + \rho_2 + \rho_3 \right] \hat{u}_p \right\} \quad (6.39)
\end{aligned}$$

where

$$\begin{aligned}\mu_1 &= \frac{1}{h_1^T \gamma_1 + m} \\ \mu_2 &= \frac{1}{h_2^T \gamma_2}\end{aligned}\tag{6.40}$$

and  $h_1^T$  and  $h_2^T$  are vectors such that:

$$\begin{aligned}h_1^T x(t) &= v_{out}(t)H(s) \\ h_2^T x(t) &= i_P(t)\end{aligned}\tag{6.41}$$

Setting  $\hat{r}_p = 1$  and  $\hat{u}_p = 0$  gives the control-to-output transfer function

$$\frac{\hat{v}_{out}}{\hat{r}}:$$

$$\begin{aligned}\frac{\hat{v}_{out}}{\hat{r}} &= \frac{\mu_1}{T_s} \left[ \left( C_2^T \eta_2 + \{ [C_3^T \eta_3 + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 \right. \right. \\ &\quad \left. \left. - \zeta_2 \mu_2 h_2^T \} \beta_2 \right) (I - \Gamma_1 \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} \xi_1 + \zeta_1 \right]\end{aligned}\tag{6.42}$$

By setting  $\hat{r}_p = 0$  and  $\hat{u}_p = 1$  the input-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{v}_g}$  can be found to be:

$$\begin{aligned}\frac{\hat{v}_{out}}{\hat{v}_g} &= \frac{1}{T_s} \left[ \left( C_2^T \eta_2 + \{ [C_3^T \eta_3 + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 - \zeta_2 \mu_2 h_2^T \} \beta_2 \right) \right. \\ &\quad \times (I - \Gamma_1 \beta_1 \beta_3 \Gamma_2 \beta_2)^{-1} \Gamma_1 [\beta_1 (\beta_3 \Gamma_2 \eta_2 B_2 + \eta_3 B_3) + \eta_1 B_1] \\ &\quad + \{ [C_3^T \eta_3 + (C_1^T \eta_1 - \zeta_1 \mu_1 h_1^T \beta_1) \beta_3] \Gamma_2 - \zeta_2 \mu_2 h_2^T \} \eta_2 B_2 + (C_1^T \eta_1 \\ &\quad \left. - \zeta_1 \mu_1 h_1^T \beta_1) \eta_3 B_3 - \zeta_1 \mu_1 h_1^T \eta_1 B_1 + \rho_1 + \rho_2 + \rho_3 \right]\end{aligned}\tag{6.43}$$

The control-to-output and input-to-output transfer functions are compared with results from simulation in Fig. 47 and Fig. 48, respectively, with the

following parameters:

$$\begin{aligned}T_s &= 10\mu s \\V_g &= 15V \\D &= 0.25 \\R_l &= 150\Omega \\m &= 0V/s\end{aligned}\tag{6.44}$$

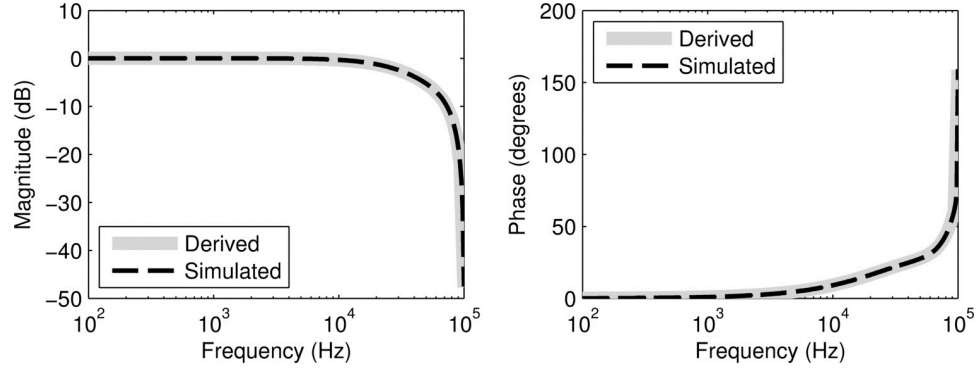


Figure 47: The derived  $V^2$  controlled boost converter in DCM control-to-output transfer function (solid) compared with results obtained from simulation (dashed).

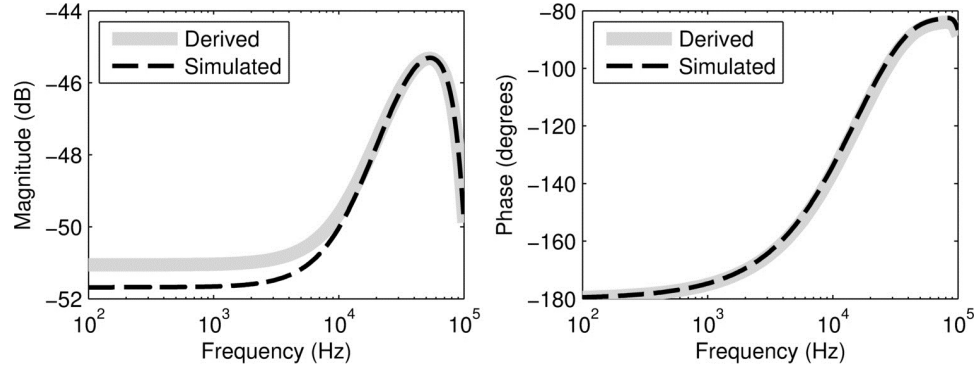


Figure 48: The derived  $V^2$  controlled boost converter in DCM input-to-output transfer function (solid) compared with results obtained from simulation (dashed).

### Section 6.3: $V^2$ Controlled Converter in DVM

As with the previously discussed control methods, a  $V^2$  controlled power converter operates in DVM when the voltage  $v_1(t)$  across the storage capacitor ( $C_1$  in the OTB converter) goes to zero, causing the switching diode  $P$  to turn on prematurely and creating a new topology before  $r(t)$  dictates the shutting off of  $Q$ . With  $N_s = 3$ ,  $T_s$  is split into three subintervals with DC lengths  $D_1T_s$ ,  $D_2T_s$ , and  $D_3T_s$  as shown in Fig. 49 which illustrates the switching period  $T_s$  for DVM described by both Model 1 and Model 2. The effective topologies of the OTB converter's power stage during the three subintervals are shown in Fig. 50. Model 2 will now be considered for the derivations.

$t_2$  is determined by the period  $T_s$  of the clock signal CLK and therefore is not sensitive to variations in  $v_g$  or  $r(t)$  such that  $\hat{t}_2 = 0$ .  $t_3$  is the time when  $v_1(t)$  reaches zero:

$$v_1(t_3) = 0 \quad (6.45)$$

The vector  $h_2^T$  is defined such that:

$$h_2^T x(t) = v_1(t) \quad (6.46)$$

allowing (6.45) to be expressed as:

$$h_2^T x(t_3) = 0 \quad (6.47)$$

Dropping the DC component of  $x(t_3)$ :

$$h_2^T \hat{x}(t_3) = 0 \quad (6.48)$$

Substituting in (2.19) and solving for  $\hat{t}_3$ :

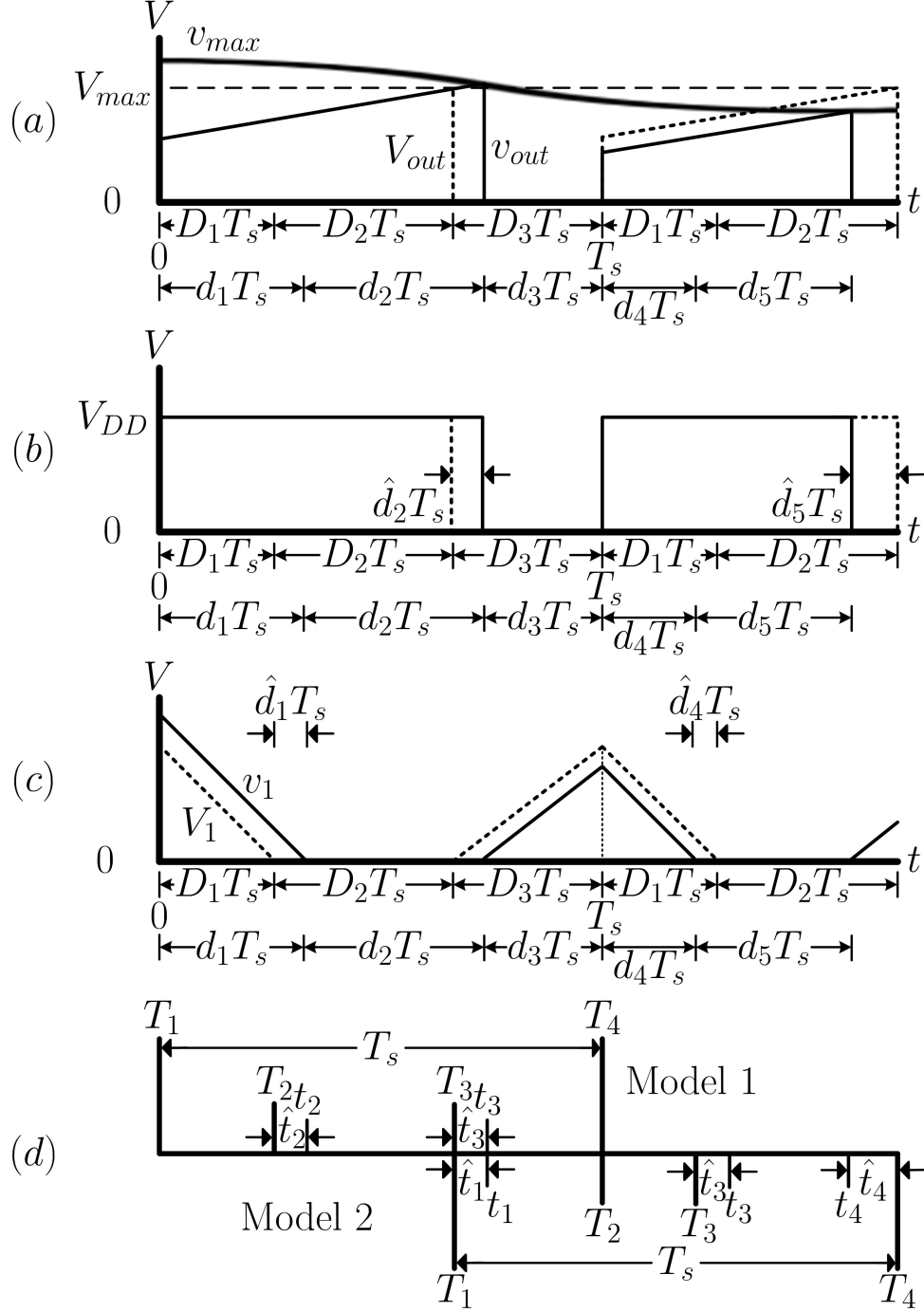


Figure 49: Switching period  $T_s$  for a  $V^2$  controlled in DVM, described for both Model 1 and Model 2. (a) shows the output voltage  $v_{out}(t)$  being compared with the peak output voltage  $v_{max}(t)$  to produce the square wave of (b). (c) shows the storage capacitor voltage  $v_1(t)$  reaching zero and creating a third intermediate topology. (d) illustrates how the times are denoted under Model 1 and Model 2.

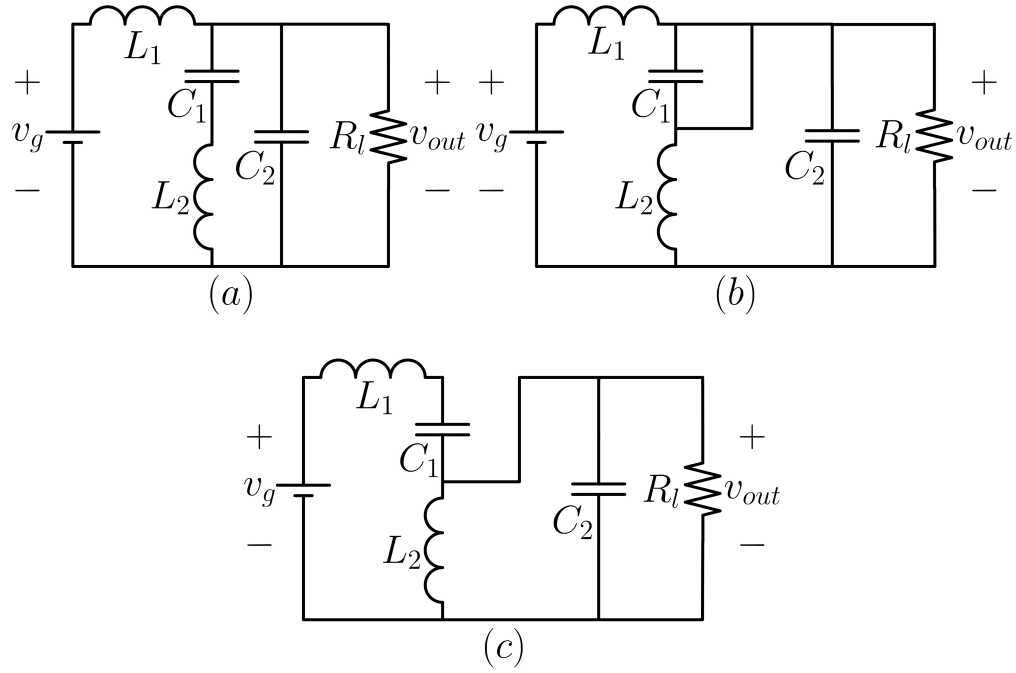


Figure 50: The effective topologies of a  $V^2$  OTB converter in DVM assuming ideal switches. (a) is the effective topology during the first subinterval of DC length  $D_1T_s$  where  $Q$  is on and  $P$  is off. (b) is the effective topology during the second subinterval of DC length  $D_2T_s$  where both  $Q$  and  $P$  are on. (c) is the effective topology during the third subinterval of DC length  $D_3T_s$  where  $Q$  is off and  $P$  is on.



$$\hat{t}_3 = -\mu_2 h_2^T \Phi_2 \hat{x}(T_2) - \mu_2 h_2^T \eta_2 B_2 \hat{u}(T_3) \quad (6.49)$$

where

$$\mu_2 = \frac{1}{h_2^T \gamma_2} \quad (6.50)$$

The expression for  $\hat{t}_4$  is the same as the expression for the  $\hat{t}_4$  of the  $V^2$  controlled converter in DCM case:

$$\hat{t}_4 = \mu_3 \hat{r}(T_4) - \mu_3 h_3^T \Phi_3 \hat{x}(T_3) - \mu_3 h_3^T \eta_3 B_3 \hat{u}(T_4) \quad (6.51)$$

where

$$\mu_3 = \frac{1}{h_3^T \gamma_3 + m} \quad (6.52)$$

$$h_3^T x(t) = v_{out}(t) H(s) \quad (6.53)$$

Finding an expression for  $\hat{x}(T_1)$  is the next step after obtaining one for  $\hat{t}_4$ . By using (2.27),  $\hat{x}(T_2)$  is:

$$\hat{x}(T_2) = \Phi_1 \hat{x}(T_1) + \eta_1 B_1 \hat{u}(T_2) \quad (6.54)$$

Using (2.27) again and substituting in (6.49) and (6.54) gives the following expression for  $\hat{x}(T_3)$ :

$$\begin{aligned} \hat{x}(T_3) &= \Phi_2 \hat{x}(T_2) + \xi_2 \hat{t}_3 + \eta_2 B_2 \hat{u}(T_3) \\ &= \Gamma_2 \Phi_2 \hat{x}(T_2) + \Gamma_2 \eta_2 B_2 \hat{u}(T_3) \\ &= \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}(T_2) + \Gamma_2 \eta_2 B_2 \hat{u}(T_3) \end{aligned} \quad (6.55)$$

where

$$\Gamma_i = I - \xi_i \mu_i h_i^T \quad (6.56)$$

Using (2.27) once more to find  $\hat{x}(T_3)$  and substituting in (6.51) and (6.55):

$$\begin{aligned} \hat{x}(T_4) &= \Phi_3 \hat{x}(T_3) + \xi_3 \hat{t}_4 + \eta_3 B_3 \hat{u}(T_4) \\ &= \Gamma_3 \Phi_3 \hat{x}(T_3) + \xi_3 \mu_3 \hat{r}(T_4) + \Gamma_3 \eta_3 B_3 \hat{u}(T_4) \\ &= \Gamma_3 \Phi_3 \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}(T_4) + \Gamma_3 \Phi_3 \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}(T_2) \\ &\quad + \Gamma_3 \Phi_3 \Gamma_2 \eta_2 B_2 \hat{u}(T_3) + \Gamma_3 \eta_3 B_3 \hat{u}(T_4) \end{aligned} \quad (6.57)$$

Expressing all of the signals in their exponential form as in (2.31) and substituting in (2.32) for  $\hat{x}(T_4)$ :

$$\begin{aligned} \hat{x}(T_1) e^{j\omega T_s} &= \Gamma_3 \Phi_3 \Gamma_2 \Phi_2 \Phi_1 \hat{x}(T_1) + \xi_3 \mu_3 \hat{r}_p e^{j\omega T_4} \\ &\quad + \Gamma_3 \Phi_3 \Gamma_2 \Phi_2 \eta_1 B_1 \hat{u}_p e^{j\omega T_2} + \Gamma_3 \Phi_3 \Gamma_2 \eta_2 B_2 \hat{u}_p e^{j\omega T_3} \\ &\quad + \Gamma_3 \eta_3 B_3 \hat{u}_p e^{j\omega T_4} \end{aligned} \quad (6.58)$$

Solving for  $\hat{x}(T_1)$ :

$$\begin{aligned} \hat{x}(T_1) &= (I - \Gamma_3 \beta_3 \Gamma_2 \beta_2 \beta_1)^{-1} \{ \xi_3 \mu_3 \hat{r}(T_1) + [\Gamma_3 \beta_3 \Gamma_2 (\beta_2 \eta_1 B_1 \\ &\quad + \eta_2 B_2) + \Gamma_3 \eta_3 B_3] \hat{u}(T_1) \} \end{aligned} \quad (6.59)$$

Now what is left is to substitute (6.51), (6.54), (6.55), and (6.59) into (3.23).

Expanding (3.23) for  $N_s = 3$ :

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} [C_1^T \eta_1 \hat{x}(T_1) e^{-j\omega T_1} + \rho_1 \hat{u}_p + C_2^T \eta_2 \hat{x}(T_2) e^{-j\omega T_2} \\
& + \zeta_2 \hat{t}_3 e^{-j\omega T_3} + \rho_2 \hat{u}_p + C_3^T \eta_3 \hat{x}(T_3) e^{-j\omega T_3} + \zeta_3 \hat{t}_4 e^{-j\omega T_4} \\
& + \rho_3 \hat{u}_p] \quad (6.60)
\end{aligned}$$

Substituting in (6.49), (6.51), (6.54), and (6.55) gives:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left( C_1^T \eta_1 + \{C_2^T \eta_2 + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 - \zeta_2 \mu_2 h_2^T] \beta_2 \} \beta_1 \right) \\
& \times \hat{x}(T_1) e^{j\omega T_1} + \zeta_3 \mu_3 \hat{r}_p + \left( \{C_2^T \eta_2 + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 \right. \\
& - \zeta_2 \mu_2 h_2^T] \beta_2 \} \eta_1 B_1 + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 - \zeta_2 \mu_2 h_2^T] \eta_2 B_2 \\
& \left. - \zeta_3 \mu_3 h_3^T \eta_3 B_3 + \rho_1 + \rho_2 + \rho_3 \right) \hat{u}_p \quad (6.61)
\end{aligned}$$

Finally substituting in (6.59):

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left\{ \left[ \left( C_1^T \eta_1 + \{C_2^T \eta_2 + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 - \zeta_2 \mu_2 h_2^T] \beta_2 \} \right. \right. \right. \\
& \times \beta_1 \Big) (I - \Gamma_3 \beta_3 \Gamma_2 \beta_2 \beta_1)^{-1} \zeta_3 \mu_3 + \zeta_3 \mu_3 \Big] \hat{r}_p + \left[ \left( C_1^T \eta_1 + \{C_2^T \eta_2 \right. \right. \\
& + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 - \zeta_2 \mu_2 h_2^T] \beta_2 \} \beta_1 \Big) (I - \Gamma_3 \beta_3 \Gamma_2 \beta_2 \beta_1)^{-1} \\
& \times (\Gamma_3 \beta_3 \Gamma_2 \beta_2 \eta_1 B_1 + \Gamma_3 \beta_3 \Gamma_2 \eta_2 B_2 + \Gamma_3 \eta_3 B_3) + \{C_2^T \eta_2 + [(C_3^T \eta_3 \\
& - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 - \zeta_2 \mu_2 h_2^T] \beta_2 \} \eta_1 B_1 + [(C_3^T \eta_3 - \zeta_3 \mu_3 h_3^T \beta_3) \Gamma_2 \\
& \left. \left. - \zeta_2 \mu_2 h_2^T] \eta_2 B_2 - \zeta_3 \mu_3 h_3^T \eta_3 B_3 + \rho_1 + \rho_2 + \rho_3 \right] \hat{u}_p \right\} \quad (6.62)
\end{aligned}$$

In DVM, rotating the subscripts back to Model 1 requires two rotations such that 1 is replaced with 3, 2 with 1, and 3 with 2:

$$\begin{aligned}
H_0(j\omega) = & \frac{1}{T_s} \left\{ \left[ \left( C_3^T \eta_3 + \{ C_1^T \eta_1 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 - \zeta_1 \mu_1 h_1^T] \beta_1 \} \right. \right. \right. \\
& \times \beta_3 \Big) (I - \Gamma_2 \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} \xi_2 \mu_2 + \zeta_2 \mu_2 \Big] \hat{r}_p + \left[ \left( C_3^T \eta_3 + \{ C_1^T \eta_1 \right. \right. \\
& + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 - \zeta_1 \mu_1 h_1^T] \beta_1 \} \beta_3 \Big) (I - \Gamma_2 \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} \\
& \times (\Gamma_2 \beta_2 \Gamma_1 \beta_1 \eta_3 B_3 + \Gamma_2 \beta_2 \Gamma_1 \eta_1 B_1 + \Gamma_2 \eta_2 B_2) + \{ C_1^T \eta_1 + [(C_2^T \eta_2 \\
& - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 - \zeta_1 \mu_1 h_1^T] \beta_1 \} \eta_3 B_3 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 \\
& \left. \left. \left. - \zeta_1 \mu_1 h_1^T] \eta_1 B_1 - \zeta_2 \mu_2 h_2^T \eta_2 B_2 + \rho_1 + \rho_2 + \rho_3 \right] \hat{u}_p \right\} \quad (6.63)
\end{aligned}$$

where

$$\begin{aligned}
\mu_1 &= \frac{1}{h_1^T \gamma_1} \\
\mu_2 &= \frac{1}{h_2^T \gamma_2 + m}
\end{aligned} \quad (6.64)$$

and  $h_1^T$  and  $h_2^T$  are vectors such that:

$$\begin{aligned}
h_1^T x(t) &= v_1(t) \\
h_2^T x(t) &= v_{out}(t) H(s)
\end{aligned} \quad (6.65)$$

By setting  $\hat{r}_p = 1$  and  $\hat{u}_p = 0$  the control-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{r}}$  can be found:

$$\begin{aligned}
\frac{\hat{v}_{out}}{\hat{r}} = & \frac{\mu_2}{T_s} \left[ \left( C_3^T \eta_3 + \{ C_1^T \eta_1 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 \right. \right. \\
& \left. \left. - \zeta_1 \mu_1 h_1^T] \beta_1 \} \beta_3 \right) (I - \Gamma_2 \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} \xi_2 + \zeta_2 \right] \quad (6.66)
\end{aligned}$$

Setting  $\hat{r}_p = 0$  and  $\hat{u}_p = 1$ , the input-to-output transfer function  $\frac{v_{out}}{\hat{v}_g}$  is:

$$\begin{aligned} \frac{\hat{v}_{out}}{\hat{v}_g} = \frac{1}{T_s} & \left[ \left( C_3^T \eta_3 + \{ C_1^T \eta_1 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 \right. \right. \\ & \quad \left. \left. - \zeta_1 \mu_1 h_1^T] \beta_1 \} \beta_3 \right) (I - \Gamma_2 \beta_2 \Gamma_1 \beta_1 \beta_3)^{-1} (\Gamma_2 \beta_2 \Gamma_1 \beta_1 \eta_3 B_3 \right. \\ & \quad \left. + \Gamma_2 \beta_2 \Gamma_1 \eta_1 B_1 + \Gamma_2 \eta_2 B_2) + \{ C_1^T \eta_1 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 \right. \\ & \quad \left. - \zeta_1 \mu_1 h_1^T] \beta_1 \} \eta_3 B_3 + [(C_2^T \eta_2 - \zeta_2 \mu_2 h_2^T \beta_2) \Gamma_1 - \zeta_1 \mu_1 h_1^T] \eta_1 B_1 \right. \\ & \quad \left. - \zeta_2 \mu_2 h_2^T \eta_2 B_2 + \rho_1 + \rho_2 + \rho_3 \right] \quad (6.67) \end{aligned}$$

The control-to-output and input-to-output transfer functions are compared with results from simulation in Fig. 51 and Fig. 52, respectively, with the following parameters:

$$\begin{aligned} T_s &= 10\mu s \\ V_g &= 28V \\ D &= 0.4 \\ R_l &= 3\Omega \\ m &= 0V/s \end{aligned} \quad (6.68)$$

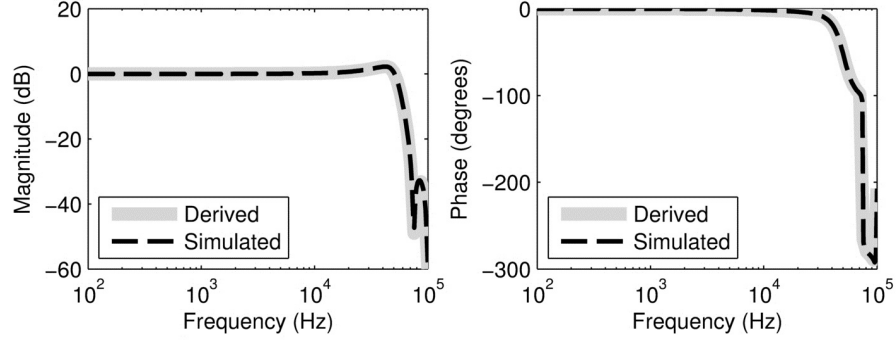


Figure 51: The derived  $V^2$  controlled OTB converter in DVM control-to-output transfer function (solid) compared with results obtained from simulation (dashed).

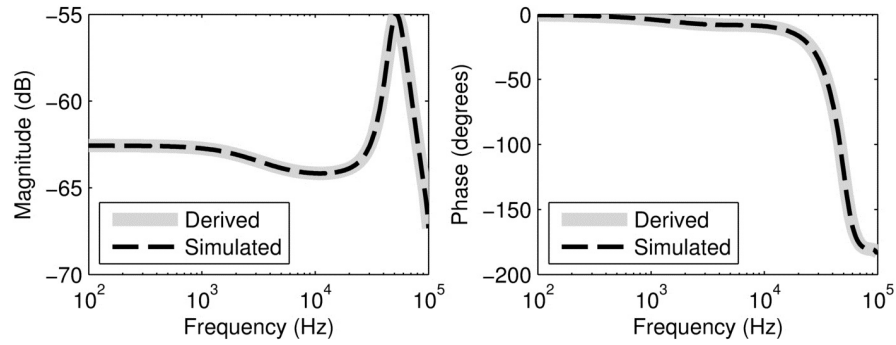


Figure 52: The derived  $V^2$  controlled OTB converter in DVM input-to-output transfer function (solid) compared with results obtained from simulation (dashed).

## Chapter 7: Verification

The transfer functions resulting from the exact method were verified by simulation, the code of which can be found in the appendix. Generally, hardware verification would have been preferred however verifying Bode plots of the control-to-output or input-to-output transfer functions requires a gain phase frequency response analyzer, an expensive instrument that is no longer available at Portland State University. When such an instrument is not available, academic papers utilize simulation engines such as SIMPLIS for verification. Unfortunately, access to SIMPLIS is also not readily available. Instead the academic paper [4] proposing the method that SIMPLIS uses for frequency response is discussed and compared to the method used by this thesis for verification. Obtaining the frequency response through simulation requires the following steps:

1. Obtain the large signal steady state of the converter with no small signal perturbation sources.
2. Obtain the small signal steady state of the converter with small signal perturbation sources turned on.
3. Compute the magnitude and phase of the sinusoidal response of the system.

The only difference between the simulation method used by this thesis and the method used by [4] which results in a change of accuracy is the small signal difference equation which is solved during step 2. By examining how the small signal difference equation used by [4] is derived, this chapter serves to show that the verification used in this thesis is both valid and an improvement on the simulation engine used by SIMPLIS.

The AC small signal difference equation was derived in Chapter 2:

$$\hat{x}(T_{i+1}) = \Phi_i \hat{x}(T_i) + \xi_i \hat{t}_{i+1} + \eta_i B_i \hat{u}(T_{i+1}) \quad (2.27)$$

The corresponding equation in [4] is:

$$\hat{x}(t_{i+1}) = \Phi_i \hat{x}(t_i) + \Psi_i \hat{u}(t_i) + \dot{X}(T_{i+1}) [\hat{t}_{i+1} - \hat{t}_i] \quad (7.1)$$

which differs from (2.27) in two ways. First, (7.1) assumes that perturbation sources are piecewise constant approximations of a sinusoidal waveform as in Fig. 53. This assumption means that  $\hat{u}(t)$  does not change from  $t_i < t < t_{i+1}$ , slightly shortening the analysis at the cost of accuracy. This assumption is not made in this thesis. Second, (7.1) uses the state vectors at switching event times  $t_i$  and  $t_{i+1}$  while (2.27) uses the DC time values  $T_i$  and  $T_{i+1}$  which is arguably more convenient but has no effect on the accuracy.

Proving that the only difference between the two methods that affects accuracy is an approximation of  $\hat{u}(t)$  requires returning to the expression for  $x(t_{i+1})$  derived previously:

$$\begin{aligned} x(t_{i+1}) = e^{A_i(t_{i+1}-t_i)} x(t_i) &+ \int_{t_i}^{t_{i+1}} e^{A_i(t_{i+1}-\tau)} B_i U d\tau \\ &+ \int_{t_i}^{t_{i+1}} e^{A_i(t_{i+1}-\tau)} B_i \hat{u} e^{j\omega\tau} d\tau \end{aligned} \quad (2.21)$$

Assuming that  $\hat{u}(t)$  is constant for  $t_i < t < t_{i+1}$  makes  $u(t_i)$  constant as well, allowing (2.21) to be adjusted:

$$x(t_{i+1}) = e^{A_i(t_{i+1}-t_i)} x(t_i) + \int_{t_i}^{t_{i+1}} e^{A_i(t_{i+1}-\tau)} B_i u(t_i) d\tau \quad (7.2)$$



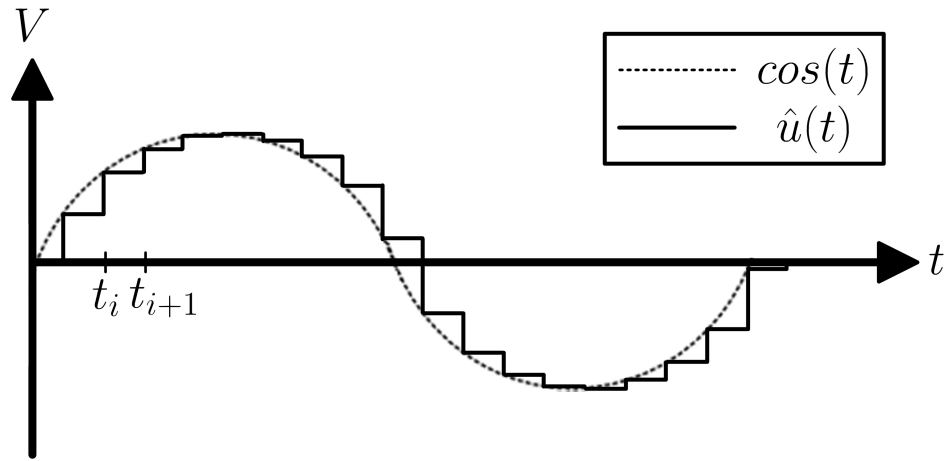


Figure 53: Approximation of a sinusoidal perturbation source  $\hat{u}(t)$  by a piecewise constant waveform. This approximation is used in simulation engines such as SIMPLIS but not made in this thesis.

Substituting  $t_i = T_i + \hat{t}_i$  into (7.2) results in:

$$\begin{aligned} x(t_{i+1}) &= e^{A_i[(T_{i+1}-T_i)+(\hat{t}_{i+1}-\hat{t}_i)]}x(t_i) + A_i^{-1}[e^{[(T_{i+1}-T_i)+(\hat{t}_{i+1}-\hat{t}_i)]} - I]B_i u(t_i) \\ &= \Phi_i e^{A_i(\hat{t}_{i+1}-\hat{t}_i)}x(t_i) + A_i^{-1}[\Phi_i e^{A_i(\hat{t}_{i+1}-\hat{t}_i)} - I]B_i u(t_i) \end{aligned} \quad (7.3)$$

Linearizing the exponents containing time perturbations  $\hat{t}_i$  and  $\hat{t}_{i+1}$  is accomplished by replacing them with the first two terms of the Taylor series expansion:

$$e^x \approx 1 + x$$

turning (7.3) into:

$$x(t_{i+1}) = \Phi_i[I + A_i(\hat{t}_{i+1} - \hat{t}_i)]x(t_i) + A_i^{-1}\Phi_i[I - A_i(\hat{t}_{i+1} - \hat{t}_i)] - IB_i u(t_i) \quad (7.4)$$

Once again, no accuracy is lost in the large signal or small signal sense as only higher order terms are neglected. Evaluating the right-hand side (7.4) and dropping any terms that are not small signal (first order) terms gives the small signal difference equation used in [4]:

$$\hat{x}(t_{i+1}) = \Phi_i \hat{x}(t_i) + \Psi_i \hat{u}(t_i) + \dot{X}(T_{i+1})[\hat{t}_{i+1} - \hat{t}_i] \quad (7.1)$$

This serves as adequate proof that (2.27) provides more accuracy than (7.1) since they are both derived the same way, with the exception of the approximation in [4] that perturbation sources are piecewise constant, and is therefore appropriate to use for verification.

## Conclusion

A procedure for obtaining the exact control-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{r}}$  and input-to-output transfer function  $\frac{\hat{v}_{out}}{\hat{u}}$  for switched DC-to-DC converters under a variety of operating modes and control methods has been given, allowing the small signal frequency behavior to be completely characterized. Doing so involved developing a mathematical representation of the switched system by describing large signal and small signal models from state space analysis. The models were then applied to the time-varying function to get a general expression containing both transfer functions of interest. The exact method has been carried out for a boost converter in NVM/CCM, NVM/DCM, CPM/CCM, CPM/DCM,  $V^2$ /CCM and  $V^2$ /DCM and for an OTB converter in NVM/DVM, CPM/DVM and  $V^2$ /DVM, the majority of which have yet to have their transfer functions published to the accuracy provided by this thesis. From these examples it can be seen that the exact method can be applied to other combinations of operating conditions and control methods by clearly defining the switching conditions and appropriately substituting them into the derived small signal difference equation of the state vector.

The resulting Bode plots of the transfer functions from the exact method match the simulations extremely well and are unparalleled in accuracy by other methods, including those used in simulators featuring frequency response such as SIMPLIS, keeping its accuracy even at very high switching frequencies. Any deviations that the derived results have from those produced by simulation can be explained by the error inherent to discrete time simulation and numerical evaluation of integrals. As seen in the code residing in the appendix, the results of this method can be easily coded and automation is certainly plausible to make it applicable to any switched linear network. From the results, which are

exact in the "small signal sense", approximations can be made with knowledge of how much accuracy is lost to yield simpler, more practical models for hand derivations that have the potential to be more accurate than any other method to date. Such a simplified model is the subject for future work.

## References

- [1] R. Tymerski, "Frequency Analysis of Time-Interval-Modulated Switched Networks", IEEE Transactions on Power Electronics, Vol. 6, No. 2, April 1991, pp. 287-295.
- [2] R. Tymerski, "Application of the Time-Varying Transfer Function for Exact Small-Signal Analysis", IEEE Transactions on Power Electronics, Vol. 9, no. 2, March 1994, pp. 196-205.
- [3] R. Tymerski, "Exact Input-to-Output Frequency Response of Time Interval Modulated Switched Networks", Power Electronics Specialists Conference, 1997. PESC '97 Record., 28th Annual IEEE, Vol.1, June 1997, pp. 428-434.
- [4] Wong, R.C. and Groves, J., "An automated small-signal frequency-domain analyzer for general periodic-operating systems as obtained via time-domain simulation," Power Electronics Specialists Conference, 1995. PESC '95 Record., 26th Annual IEEE , vol.2, pp.801-808
- [5] Tymerski, R. and Vorperian, V., "Generation and classification of PWM DC-to-DC converters," Aerospace and Electronic Systems, IEEE Transactions on , vol.24, no.6, Nov. 1988, pp.743-754

## Appendix: MATLAB Code

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% dp_interval.m
% fzero returns length of second interval for dcm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [htx]=dp_interval(dp,a1,b1,a2,b2,a3,b3,ts,d,vg,ht);
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,dp*ts);
[phi3,psi3]=c2d(a3,b3,(1-d-dp)*ts);
x1=(I-phi3*phi2*phi1)\(phi3*phi2*psi1+phi3*psi2+psi3)*vg;
x3=phi2*phi1*x1+(phi2*psi1+psi2)*vg;
htx=ht*x3;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% dvm_interval.m
% fzero returns length of second subinterval in DVM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [htx]=dvm_interval(dv,a1,b1,a2,b2,a3,b3,ts,d,vg,ht)
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,dv*ts);
[phi2,psi2]=c2d(a2,b2,(d-dv)*ts);
[phi3,psi3]=c2d(a3,b3,(1-d)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
htx=ht*(x3);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% loadCCM.m
% loads NVM boost CCM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lf=58e-6;cf=5.5e-6;r=18.6;vg=15;d=0.25;ts=1/100e3;
a1=[0 0;0 -1/(r*cf)];
a2=[0 -1/lf;1/cf -1/(r*cf)];
b1=[1/lf; 0];
b2=[1/lf; 0];
c1=[0 1];
c2=[0 1];
e1=0;
e2=0;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CCMexact.m
% NVM boost CCM exact results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadCCM;
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,(1-d)*ts);
x2=(I-phi1*phi2)\(phi1*psi2+psi1)*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
zeta1=(c1-c2)*x2+(e1-e2)*vg;

k=300;
f=logspace(2,5,k);
w=2*pi*f;
for n=1:k
    beta1=exp(-1i*w(n)*d*ts)*phi1;
    beta2=exp(-1i*w(n)*(1-d)*ts)*phi2;
    chi1=(1i*w(n)*I-a1);
    chi2=(1i*w(n)*I-a2);
    eta1=chi1\ (I-beta1);
    eta2=chi2\ (I-beta2);

    co1=c1*eta1*beta2+c2*eta2;
    co2=(I-beta1*beta2)\xi1;
    co(n)=co1*co2+zeta1;

    io1=(c2*eta2+c1*eta1*beta2);
    io2=(I-beta1*beta2)\(beta1*eta2*b2+eta1*b1);
    io3=c1*eta1*eta2*b2;
    io4=c1*(chi1\ (d*ts*I-eta1))*b1+e1*d*ts;
    io5=c2*(chi2\ ((1-d)*ts*I-eta2))*b2+e2*(1-d)*ts;
    io(n)=(1/ts)*(io1*io2+io3+io4+io5);
end

comag=20*log10(abs(co));
cophase=180/pi*unwrap(angle(co));
iomag=20*log10(abs(io));
iophase=180/pi*unwrap(angle(io));

hold off
semilogx(f,comag)
pause

```

```

semilogx(f,cophase)
pause
semilogx(f,iomag)
pause
semilogx(f,iophase)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CCMsimco.m
% NVM boost CCM simulation control-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadCCM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,(1-d)*ts);
x2=(I-phi1*phi2)\(phi1*psi2+psi1)*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);

for m=1:length(fac)
    beta1=phi1*exp(-1i*wac(m)*d*ts);
    beta2=phi2*exp(-1i*wac(m)*(1-d)*ts);
    phasor=(I-beta2*beta1)\beta2*xi1*ts;
    t(1)=0;
    x=phasor;
    y(1)=c1*x;
    n=2;
    while t(n-1)+tstep < d*ts;
        t(n)=t(n-1)+tstep;
        x=phi1d*x;
        y(n)=c1*x;
        n=n+1;
    end
    phid=expm(a1*(d*ts-t(n-1)));
    t(n)=d*ts;

```



```

x=phid*x+xi1*exp(1i*wac(m)*d*ts)*ts;
y(n)=c1*x;
n=n+1;
while t(n-1)+tstep < ts;
t(n)=t(n-1)+tstep;
x=phi2d*x;
y(n)=c2*x;
n=n+1;
end
phiend=expm(a2*(ts-t(n-1)));
t(n)=ts;
x=phiend*x;
y(n)=c2*x;
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=1/ts*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CCMsimio.m
% NVM boost CCM simulation input-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadCCM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

phi1=expm(a1*d*ts);
phi2=expm(a2*(1-d)*ts);

I=eye(size(a1));

```

```

tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*d*ts);
beta2=phi2*exp(-1i*wac(m)*(1-d)*ts);
chi1=1i*wac(m)*I-a1;
chi2=1i*wac(m)*I-a2;
eta1=chi1\ (I-beta1);
eta2=chi2\ (I-beta2);
phasor=(I-beta2*beta1)\ (beta2*eta1*b1+eta2*b2);
beta1d=phi1d*exp(-1i*wac(m)*tstep);
eta1d=chi1\ (I-beta1d);
beta2d=phi2d*exp(-1i*wac(m)*tstep);
eta2d=chi2\ (I-beta2d);
t(1)=0;
x=phasor;
y(1)=c1*x+e1;
n=2;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x+eta1d*b1*exp(1i*wac(m)*t(n));
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
end
phid=expm(a1*(d*ts-t(n-1)));
betad=phid*exp(-1i*wac(m)*(d*ts-t(n-1)));
etad=chi1\ (I-betad);
t(n)=d*ts;
x=phid*x+etad*b1*exp(1i*wac(m)*d*ts);
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < ts;
t(n)=t(n-1)+tstep;
x=phi2d*x+eta2d*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
end
phiend=expm(a2*(ts-t(n-1)));
betaend=phiend*exp(-1i*wac(m)*(ts-t(n-1)));
etaend=chi2\ (I-betaend);
t(n)=ts;
x=phiend*x+etaend*b2*exp(1i*wac(m)*ts);

```

```

y(n)=c2*x+e2*exp(1i*wac(m)*ts);
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=(1/ts)*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg)
pause
semilogx(fac,ph)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% loadDCM.m
% loads NVM boost DCM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lf=58e-6;cf=5.5e-6;r=150;vg=15;d=0.25;ts=1/100e3;
a1=[0 0;0 -1/(r*cf)];
a2=[0 -1/lf;1/cf -1/(r*cf)];
a3=[0 0;0 -1/(r*cf)];
b1=[1/lf; 0];
b2=[1/lf; 0];
b3=[0; 0];
c1=[0 1];
c2=[0 1];
c3=[0 1];
e1=0;
e2=0;
e3=0;
ht=[1 0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DCMexact.m
% NVM boost DCM exact results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadDCM;
I=eye(size(a1));
dp=fzero(@(x)dp_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht),(1-d)/2);
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,dp*ts);
[phi3,psi3]=c2d(a3,b3,(1-d-dp)*ts);

```

```

x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
zeta1=(c1-c2)*x2+(e1-e2)*vg;
zeta2=(c2-c3)*x3+(e2-e3)*vg;
gamma2=a2*x3+b2*vg;
mu2=1/(ht*gamma2);
Gamma2=I-xi2*ht*mu2;

k=100;
f=logspace(2,5,k);
w=2*pi*f;
for n=1:k
beta1=exp(-1i*w(n)*d*ts)*phi1;
beta2=exp(-1i*w(n)*dp*ts)*phi2;
beta3=exp(-1i*w(n)*(1-d-dp)*ts)*phi3;
chi1=(1i*w(n)*I-a1);
chi2=(1i*w(n)*I-a2);
chi3=(1i*w(n)*I-a3);
eta1=chi1\ (I-beta1);
eta2=chi2\ (I-beta2);
eta3=chi3\ (I-beta3);

co1=c2*eta2-zeta2*ht*mu2*beta2;
co2=c3*eta3*Gamma2*beta2+c1*eta1*beta3*Gamma2*beta2;
co3=(I-beta1*beta3*Gamma2*beta2)\xi1;
co(n)=(co1+co2)*co3+zeta1;

rho1=c1*(chi1\ (d*ts*I-eta1))*b1+e1*d*ts;
rho2=c2*(chi2\ (dp*ts*I-eta2))*b2+e2*dp*ts;
rho3=c3*(chi3\ ((1-d-dp)*ts*I-eta3))*b3+e3*(1-d-dp)*ts;

io1=c1*eta1*beta3*Gamma2*beta2+c2*eta2;
io2=c3*eta3*Gamma2*beta2-zeta2*ht*mu2*beta2;
io3=I-beta1*beta3*Gamma2*beta2;
io4=eta1*b1+beta1*beta3*Gamma2*eta2*b2+beta1*eta3*b3;
io5=c1*eta1*beta3*Gamma2*eta2*b2+c1*eta1*eta3*b3;
io6=c3*eta3*Gamma2*eta2*b2-zeta2*ht*mu2*eta2*b2;
io(n)=(1/ts)*((io1+io2)*(io3\*io4)+io5+io6+rho1+rho2+rho3);
end

comag=20*log10(abs(co));
cophase=180/pi*unwrap(angle(co));

```

```

iomag=20*log10(abs(io));
iophase=180/pi*unwrap(angle(io));

hold off
semilogx(f,comag)
pause
semilogx(f,cophase)
pause
semilogx(f,iomag)
pause
semilogx(f,iophase)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DCMsimco.m
% NVM boost DCM simulation control-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadDCM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

dp=fzero(@(x)dp_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht),(1-d)/2);
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,dp*ts);
[phi3,psi3]=c2d(a3,b3,(1-d-dp)*ts);
x2=(I-phi1*phi3*psi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;

I=eye(size(a1));
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
mu2=1/(ht*(a2*x3+b2*vg));
Gamma2=I-xi2*ht*mu2;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*d*ts);

```

```

beta2=phi2*exp(-1i*wac(m)*dp*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d-dp)*ts);
phasor=(I-beta3*Gamma2*beta2*beta1)\beta3*Gamma2*beta2*xi1*ts;

t(1)=0;
x=phasor;
y(1)=c1*x;
n=2;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x;
y(n)=c1*x;
n=n+1;
end
phid=expm(a1*(d*ts-t(n-1)));
t(n)=d*ts;
x=phid*x+xi1*exp(1i*wac(m)*d*ts)*ts;
y(n)=c1*x;
n=n+1;
while t(n-1)+tstep < (d+dp)*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x;
y(n)=c2*x;
n=n+1;
end
phidp=expm(a2*((d+dp)*ts-t(n-1)));
t(n)=(d+dp)*ts;
x=Gamma2*phidp*x;
y(n)=c2*x;
n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x;
y(n)=c3*x;
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
t(n)=ts;
x=phiend*x;
y(n)=c3*x;
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end

```

```

z(m)=1/ts*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DCMsimio.m
% NVM boost DCM simulation input-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadDCM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

dp=fzero(@(x)dp_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht),(1-d)/2);
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,dp*ts);
[phi3,psi3]=c2d(a3,b3,(1-d-dp)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
mu2=1/(ht*(a2*x3+b2*vg));
Gamma2=I-xi2*ht*mu2;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*d*ts);
beta2=phi2*exp(-1i*wac(m)*dp*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d-dp)*ts);
chi1=1i*wac(m)*I-a1;

```

```

chi2=1i*wac(m)*I-a2;
chi3=1i*wac(m)*I-a3;
eta1=chi1/(I-beta1);
eta2=chi2/(I-beta2);
eta3=chi3/(I-beta3);
phasor1=(I-beta3*Gamma2*beta2*beta1);
phasor2=(beta3*Gamma2*(beta2*eta1*b1+eta2*b2)+eta3*b3);
phasor=phasor1/phasor2;

beta1d=phi1d*exp(-1i*wac(m)*tstep);
eta1d=chi1/(I-beta1d);
beta2d=phi2d*exp(-1i*wac(m)*tstep);
eta2d=chi2/(I-beta2d);
beta3d=phi3d*exp(-1i*wac(m)*tstep);
eta3d=chi3/(I-beta3d);
t(1)=0;
x=phasor;
y(1)=c1*x+e1;
n=2;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x+eta1d*b1*exp(1i*wac(m)*t(n));
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
end
phid=expm(a1*(d*ts-t(n-1)));
betad=phid*exp(-1i*wac(m)*(d*ts-t(n-1)));
etad=chi1/(I-betad);
t(n)=d*ts;
x=phid*x+etad*b1*exp(1i*wac(m)*d*ts);
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < (d+dp)*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x+eta2d*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
end
phidp=expm(a2*((d+dp)*ts-t(n-1)));
betadp=phidp*exp(-1i*wac(m)*((d+dp)*ts-t(n-1)));
etadp=chi2/(I-betadp);
t(n)=(d+dp)*ts;
x=Gamma2*phidp*x+Gamma2*etadp*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));

```



```

n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x+eta3d*b3*exp(1i*wac(m)*t(n));
y(n)=c3*x+e3*exp(1i*wac(m)*t(n));
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
betaend=phiend*exp(-1i*wac(m)*(ts-t(n-1)));
etaend=chi3\*(I-betaend);
t(n)=ts;
x=phiend*x+etaend*b3*exp(1i*wac(m)*ts);
y(n)=c3*x+e3*exp(1i*wac(m)*ts);
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=(1/ts)*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% loadDVM.m
% loads NVM OTB DVM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
vg=28;r=3;l1=130e-6;l2=1e-3;c1=60e-9;c2=50e-6;d=0.4;ts=1/100e3;
a1=[0 0 0 -1/l1;
0 0 -1/l2 1/l2;
0 1/c1 0 0;
1/c2 -1/c2 0 -1/(r*c2)];
a2=[0 0 0 -1/l1;
0 0 0 1/l2;
0 0 0 0;
1/c2 -1/c2 0 -1/(r*c2)];
a3=[0 0 -1/l1 -1/l1;
0 0 0 1/l2;
1/c1 0 0 0;
1/c2 -1/c2 0 -1/(r*c2)];

```

```

b1=[1/l1; 0; 0; 0];
b2=[1/l1; 0; 0; 0];
b3=[1/l1; 0; 0; 0];
c1=[0 0 0 1];
c2=[0 0 0 1];
c3=[0 0 0 1];
e1=0;
e2=0;
e3=0;
ht=[0 0 1 0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DVMexact.m
% NVM OTB DVM exact results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadDVM;
dv=fzero(@(x)dvm_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht),d/2);
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,dv*ts);
[phi2,psi2]=c2d(a2,b2,(d-dv)*ts);
[phi3,psi3]=c2d(a3,b3,(1-d)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
zeta1=(c1-c2)*x2+(e1-e2)*vg;
zeta2=(c2-c3)*x3+(e2-e3)*vg;
mu1=1/(ht*(a1*x2+b1*vg));
Gamma1=I-xi1*ht*mu1;

k=300;
f=logspace(2,5,k);
w=2*pi*f;
for n=1:k
    beta1=exp(-1i*w(n)*dv*ts)*phi1;
    beta2=exp(-1i*w(n)*(d-dv)*ts)*phi2;
    beta3=exp(-1i*w(n)*(1-d)*ts)*phi3;
    chi1=(1i*w(n)*I-a1);
    chi2=(1i*w(n)*I-a2);
    chi3=(1i*w(n)*I-a3);
    eta1=chi1\ (I-beta1);
    eta2=chi2\ (I-beta2);
    eta3=chi3\ (I-beta3);

```

```

co1=(c1*eta1-zeta1*mu1*ht*beta1)*beta3;
co2=c2*eta2*Gamma1*beta1*beta3+c3*eta3;
co3=(I-beta2*Gamma1*beta1*beta3)\xi2;
co(n)=(co1+co2)*co3+zeta2;

rho1=c1*(chi1\((dv*ts*I-eta1))*b1+e1*dv*ts;
rho2=c2*(chi2\((d-dv)*ts*I-eta2))*b2+e2*(d-dv)*ts;
rho3=c3*(chi3\((1-d)*ts*I-eta3))*b3+e3*(1-d)*ts;

io1=(c1*eta1-zeta1*mu1*ht*beta1)*beta3;
io2=c2*eta2*Gamma1*beta1*beta3+c3*eta3;
io3=I-beta2*Gamma1*beta1*beta3;
io4=beta2*Gamma1*beta1*eta3*b3+beta2*Gamma1*eta1*b1+eta2*b2;

io51=c1*eta1*eta3*b3;
io52=zeta1*mu1*ht*beta1*eta3*b3;
io53=zeta1*mu1*ht*eta1*b1;
io5=io51-io52-io53;

io6=c2*eta2*Gamma1*beta1*eta3*b3+c2*eta2*Gamma1*eta1*b1;
io(n)=(1/ts)*((io1+io2)*(io3\io4)+io5+io6+rho1+rho2+rho3);
end

comag=20*log10(abs(co));
cophase=180/pi*unwrap(angle(co));
iomag=20*log10(abs(io));
iophase=180/pi*unwrap(angle(io));

hold off
semilogx(f,comag);
pause
semilogx(f,cophase);
pause
semilogx(f,iomag);
pause
semilogx(f,iophase);

%%%%%%%%%%%%%%
% DVMsimco.m
% NVM OTB DVM simulation control-to-output
%%%%%%%%%%%%%%
loadDVM;
lowfreq=100;
highfreq=100000;

```

```

Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

I=eye(size(a1));
dv=fzero(@(x)dvm_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht),d/2);
[phi1,psi1]=c2d(a1,b1,dv*ts);
[phi2,psi2]=c2d(a2,b2,(d-dv)*ts);
[phi3,psi3]=c2d(a3,b3,(1-d)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;

I=eye(size(a1));
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
mu1=1/(ht*(a1*x2+b1*vg));
Gamma1=I-xi1*ht*mu1;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*dv*ts);
beta2=phi2*exp(-1i*wac(m)*(d-dv)*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d)*ts);
phasor=(I-beta3*beta2*Gamma1*beta1)\beta3*xi2*ts;

t(1)=0;
x=phasor;
y(1)=c1*x;
n=2;
while t(n-1)+tstep < dv*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x;
y(n)=c1*x;
n=n+1;
end
phidv=expm(a1*(dv*ts-t(n-1)));
t(n)=dv*ts;
x=Gamma1*phidv*x;
y(n)=c1*x;

```

```

n=n+1;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x;
y(n)=c2*x;
n=n+1;
end
phid=expm(a2*(d*ts-t(n-1)));
t(n)=d*ts;
x=phid*x+xi2*exp(1i*wac(m)*d*ts)*ts;
y(n)=c2*x;
n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x;
y(n)=c3*x;
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
t(n)=ts;
x=phiend*x;
y(n)=c3*x;
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=1/ts*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DVMsimio.m
% NVM OTB DVM simulation input-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadDVM;
lowfreq=100;
highfreq=100000;
Ms=100;

```

```

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

I=eye(size(a1));
dv=fzero(@(x)dvm_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht),d/2);
[phi1,psi1]=c2d(a1,b1,dv*ts);
[phi2,psi2]=c2d(a2,b2,(d-dv)*ts);
[phi3,psi3]=c2d(a3,b3,(1-d)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1 = (a1-a2)*x2+(b1-b2)*vg;
xi2 = (a2-a3)*x3+(b2-b3)*vg;

I=eye(size(a1));
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
mu1=1/(ht*(a1*x2+b1*vg));
Gamma1=I-xi1*ht*mu1;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*dv*ts);
beta2=phi2*exp(-1i*wac(m)*(d-dv)*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d)*ts);
chi1=1i*wac(m)*I-a1;
chi2=1i*wac(m)*I-a2;
chi3=1i*wac(m)*I-a3;
eta1=chi1\ (I-beta1);
eta2=chi2\ (I-beta2);
eta3=chi3\ (I-beta3);
phasor1=I-beta3*beta2*Gamma1*beta1;
phasor2=beta3*beta2*Gamma1*eta1*b1+beta3*eta2*b2+eta3*b3;
phasor=phasor1\phasor2;
beta1d=phi1d*exp(-1i*wac(m)*tstep);
eta1d=chi1\ (I-beta1d);
beta2d=phi2d*exp(-1i*wac(m)*tstep);
eta2d=chi2\ (I-beta2d);
beta3d=phi3d*exp(-1i*wac(m)*tstep);
eta3d=chi3\ (I-beta3d);
t(1)=0;
x=phasor;

```

```

y(1)=c1*x+e1;
n=2;
while t(n-1)+tstep < dv*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x+eta1d*b1*exp(1i*wac(m)*t(n));
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
end
phid=expm(a1*(dv*ts-t(n-1)));
betad=phid*exp(-1i*wac(m)*(dv*ts-t(n-1)));
etad=chi1/(I-betad);
t(n)=dv*ts;
x=Gamma1*phid*x+Gamma1*etad*b1*exp(1i*wac(m)*dv*ts);
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x+eta2d*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
end
phidp=expm(a2*(d*ts-t(n-1)));
betadp=phidp*exp(-1i*wac(m)*(d*ts-t(n-1)));
etadp=chi2/(I-betadp);
t(n)=d*ts;
x=phidp*x+etadp*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x+eta3d*b3*exp(1i*wac(m)*t(n));
y(n)=c3*x+e3*exp(1i*wac(m)*t(n));
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
betaend=phiend*exp(-1i*wac(m)*(ts-t(n-1)));
etaend=chi3/(I-betaend);
t(n)=ts;
x=phiend*x+etaend*b3*exp(1i*wac(m)*ts);
y(n)=c3*x+e3*exp(1i*wac(m)*ts);
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end

```

```

z(m)=(1/ts)*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% loadCPMCCM.m
% loads CPM boost CCM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lf=58e-6;cf=5.5e-6;r=18.6;rs=0.5;vg=15;d=0.25;ts=1/100e3;
a1=[-rs/lf 0;0 -1/(r*cf)];
a2=[0 -1/lf;1/cf -1/(r*cf)];
b1=[1/lf; 0];
b2=[1/lf; 0];
c1=[0 1];
c2=[0 1];
e1=0;
e2=0;
I=eye(2);
ht=[1 0];
m=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CPMCCMexact.m
% CPM boost CCM exact results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadCPMCCM;
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,(1-d)*ts);
x2=(I-phi1*phi2)\(phi1*psi2+psi1)*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
zeta1=(c1-c2)*x2+(e1-e2)*vg;
gamma1=a1*x2+b1*vg;
mu1=1/(ht*gamma1+m);
Gamma1=I-xi1*ht*mu1;

k=10000;
f=logspace(2,5,k);

```



```

w=2*pi*f;
for n=1:k
    beta1=exp(-1i*w(n)*d*ts)*phi1;
    beta2=exp(-1i*w(n)*(1-d)*ts)*phi2;
    chi1=(1i*w(n)*I-a1);
    chi2=(1i*w(n)*I-a2);
    eta1=chi1/(I-beta1);
    eta2=chi2/(I-beta2);

    co1=c1*eta1*beta2+c2*eta2-zeta1*ht*mu1*beta1*beta2;
    co2=(I-Gamma1*beta1*beta2)/xi1;
    co(n)=(mu1/ts)*(co1*co2+zeta1);

    rho1=c1*(chi1/(d*ts*I-eta1))*b1+e1*d*ts;
    rho2=c2*(chi2/((1-d)*ts*I-eta2))*b2+e2*(1-d)*ts;

    io1=c1*eta1*beta2+c2*eta2-zeta1*mu1*ht*beta1*beta2;
    io2=I-Gamma1*beta1*beta2;
    io3=Gamma1*beta1*eta2*b2+Gamma1*eta1*b1;
    io4=c1*eta1*eta2*b2-zeta1*mu1*ht*(eta1*b1+beta1*eta2*b2);
    io(n)=(1/ts)*(io1*(io2/io3)+io4+rho1+rho2);
end

comag=20*log10(abs(co));
cophase=180/pi*unwrap(angle(co));
iomag=20*log10(abs(io));
iophase=180/pi*unwrap(angle(io));

hold off
semilogx(f,comag)
pause
semilogx(f,cophase)
pause
semilogx(f,iomag)
pause
semilogx(f,iophase)

%%%%%%%%%%%%%%
% CPMCCMsimco.m
% CPM boost CCM simulation control-to-output
%%%%%%%%%%%%%%
loadCPMCCM;
lowfreq=100;
highfreq=100000;

```

```

Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,(1-d)*ts);
x2=(I-phi1*phi2)\(phi1*psi2+psi1)*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;

tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
mu1=1/(ht*(a1*x2+b1*vg));
Gamma1=I-xi1*ht*mu1;

for m=1:length(fac)
    beta1=phi1*exp(-1i*wac(m)*d*ts);
    beta2=phi2*exp(-1i*wac(m)*(1-d)*ts);
    phasor=(I-beta2*Gamma1*beta1)\beta2*xi1*mu1;

    t(1)=0;
    x=phasor;
    y(1)=c1*x;
    n=2;
    while t(n-1)+tstep < d*ts;
        t(n)=t(n-1)+tstep;
        x=phi1d*x;
        y(n)=c1*x;
        n=n+1;
    end
    phid=expm(a1*(d*ts-t(n-1)));
    t(n)=d*ts;
    x=Gamma1*phid*x+xi1*mu1*exp(1i*wac(m)*d*ts);
    y(n)=c1*x;
    n=n+1;
    while t(n-1)+tstep < ts;
        t(n)=t(n-1)+tstep;
        x=phi2d*x;
        y(n)=c2*x;
        n=n+1;
    end
end

```

```

phiend=expm(a2*(ts-t(n-1)));
t(n)=ts;
x=phiend*x;
y(n)=c2*x;
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=1/ts*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CPMCCMsimio.m
% CPM boost CCM simulation input-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadCPMCCM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac = 2*pi*fac;

I = eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,(1-d)*ts);
x2=(I-phi1*phi2)\(phi1*psi2+psi1)*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
mu1=1/(ht*(a1*x2+b1*vg));
Gamma1=I-xi1*ht*mu1;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*d*ts);

```

```

beta2=phi2*exp(-1i*wac(m)*(1-d)*ts);
chi1=1i*wac(m)*I-a1;
chi2=1i*wac(m)*I-a2;
eta1=chi1/(I-beta1);
eta2=chi2/(I-beta2);
phasor1=(I-beta2*Gamma1*beta1)
phasor2=(beta2*Gamma1*eta1*b1+eta2*b2);
phasor=phasor1/phasor2;

beta1d=phi1d*exp(-1i*wac(m)*tstep);
eta1d=chi1/(I-beta1d);
beta2d=phi2d*exp(-1i*wac(m)*tstep);
eta2d=chi2/(I-beta2d);

t(1)=0;
x=phasor;
y(1)=c1*x+e1;
n=2;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x+eta1d*b1*exp(1i*wac(m)*t(n));
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
end
phid=expm(a1*(d*ts-t(n-1)));
betad=phid*exp(-1i*wac(m)*(d*ts-t(n-1)));
etad=chi1/(I-betad);
t(n)=d*ts;
x=Gamma1*phid*x+Gamma1*etad*b1*exp(1i*wac(m)*d*ts);
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < ts;
t(n)=t(n-1)+tstep;
x=phi2d*x+eta2d*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
end
phiend=expm(a2*(ts-t(n-1)));
betaend=phiend*exp(-1i*wac(m)*(ts-t(n-1)));
etaend=chi2/(I-betaend);
t(n)=ts;
x=phiend*x+etaend*b2*exp(1i*wac(m)*ts);
y(n)=c2*x+e2*exp(1i*wac(m)*ts);
n=n+1;

```

```

for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=(1/ts)*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg)
pause
semilogx(fac,ph)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% loadCPMDCM.m
% loads CPM boost DCM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lf=58e-6;cf=5.5e-6;r=150;rs=0.5;vg=15;d=0.25;ts=1/100e3;
a1=[-rs/lf 0;0 -1/(r*cf)];
a2=[0 -1/lf;1/cf -1/(r*cf)];
a3=[0 0;0 -1/(r*cf)];
b1=[1/lf; 0];
b2=[1/lf; 0];
b3=[0; 0];
c1=[0 1];
c2=[0 1];
c3=[0 1];
e1=0;
e2=0;
e3=0;
ht1=[1 0];
ht2=[1 0];
m=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CPMDCMexact.m
% CPM boost DCM exact results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadCPMDCM;
dp=fzero(@(x)dp_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht2),(1-d)/2);
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,dp*ts);
[phi3,psi3]=c2d(a3,b3,(1-d-dp)*ts);

```

```

x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
zeta1=(c1-c2)*x2+(e1-e2)*vg;
zeta2=(c2-c3)*x3+(e2-e3)*vg;
m=0;
gamma1=a1*x2+b1*vg;
gamma2=a2*x3+b2*vg;
mu1=1/(ht1*gamma1+m);
mu2=1/(ht2*gamma2);
Gamma1=I-xi1*ht1*mu1;
Gamma2=I-xi2*ht2*mu2;

k=100;
f=logspace(2,5,k);
w=2*pi*f;
for n=1:k
beta1=exp(-1i*w(n)*d*ts)*phi1;
beta2=exp(-1i*w(n)*dp*ts)*phi2;
beta3=exp(-1i*w(n)*(1-d-dp)*ts)*phi3;
chi1=(1i*w(n)*I-a1);
chi2=(1i*w(n)*I-a2);
chi3=(1i*w(n)*I-a3);
eta1=chi1\ (I-beta1);
eta2=chi2\ (I-beta2);
eta3=chi3\ (I-beta3);

co1=(c1*eta1-zeta1*mu1*ht1*beta1)*beta3*Gamma2*beta2;
co2=c2*eta2-zeta2*mu2*ht2*beta2+c3*eta3*Gamma2*beta2;
co3=I-Gamma1*beta1*beta3*Gamma2*beta2;
co(n)=(mu1/ts)*((co1+co2)*(co3\xi1)+zeta1);

rho1=c1*(chi1\ (d*ts*I-eta1))*b1+e1*d*ts;
rho2=c2*(chi2\ (dp*ts*I-eta2))*b2+e2*dp*ts;
rho3=c3*(chi3\ ((1-d-dp)*ts*I-eta3))*b3+e3*(1-d-dp)*ts;

io1=c2*eta2-zeta2*mu2*ht2*beta2+c3*eta3*Gamma2*beta2;
io2=(c1*eta1-zeta1*mu1*ht1*beta1)*beta3*Gamma2*beta2;
io3=I-Gamma1*beta1*beta3*Gamma2*beta2;

io41=beta1*beta3*Gamma2*eta2*b2;
io42=beta1*eta3*b3+eta1*b1;
io4=Gamma1*(io41+io42);

```

```

io5=(c3*eta3+c1*eta1*beta3)*Gamma2*eta2*b2+c1*eta1*eta3*b3;

io61=zeta2*mu2*ht2*eta2*b2;
io62=zeta2*mu1*ht1*beta1*beta3*Gamma2*eta2*b2;
io6=io61+io62;

io7=zeta1*mu1*ht1*beta1*eta3*b3+zeta1*mu1*ht1*eta1*b1;
io(n)=(1/ts)*((io1+io2)*(io3\io4)+io5-io6-io7+rho1+rho2+rho3);
end

comag=20*log10(abs(co));
cophase=180/pi*unwrap(angle(co));
iomag=20*log10(abs(io));
iophase=180/pi*unwrap(angle(io));

hold off
semilogx(f,comag)
pause
semilogx(f,cophase)
pause
semilogx(f,iomag)
pause
semilogx(f,iophase)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CPMDCMsimco.m
% CPM boost DCM simulation control-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadCPMDCM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

dp=fzero(@(x)dp_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht2),(1-d)/2);
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,dp*ts);
[phi3,psi3]=c2d(a3,b3,(1-d-dp)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;

```

```

x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
mu1=1/(ht1*(a1*x2+b1*vg)+m);
Gamma1=I-xi1*mu1*ht1;
mu2=1/(ht2*(a2*x3+b2*vg));
Gamma2=I-xi2*mu2*ht2;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*d*ts);
beta2=phi2*exp(-1i*wac(m)*dp*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d-dp)*ts);
phasor1=(I-beta3*Gamma2*beta2*Gamma1*beta1)
phasor2=beta3*Gamma2*beta2*xi1*mu1;
phasor=phasor1\phasor2;

t(1)=0;
x=phasor;
y(1)=c1*x;
n=2;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x;
y(n)=c1*x;
n=n+1;
end
phid=expm(a1*(d*ts-t(n-1)));
t(n)=d*ts;
x=Gamma1*phid*x+xi1*mu1*exp(1i*wac(m)*d*ts);
y(n)=c1*x;
n=n+1;
while t(n-1)+tstep < (d+dp)*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x;
y(n)=c2*x;
n=n+1;
end
phidp=expm(a2*((d+dp)*ts-t(n-1)));
t(n)=(d+dp)*ts;
x=Gamma2*phidp*x;

```



```

y(n)=c2*x;
n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x;
y(n)=c3*x;
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
t(n)=ts;
x=phiend*x;
y(n)=c3*x;
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=1/ts*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CPMDCMsimio.m
% CPM boost DCM simulation input-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadCPMDCM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

dp=fzero(@(x)dp_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht2),(1-d)/2);
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,dp*ts);
[phi3,psi3]=c2d(a3,b3,(1-d-dp)*ts);

```

```

x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
mu1=1/(ht1*(a1*x2+b1*vg)+m);
Gamma1=I-xi1*mu1*ht1;
mu2=1/(ht2*(a2*x3+b2*vg));
Gamma2=I-xi2*mu2*ht2;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*d*ts);
beta2=phi2*exp(-1i*wac(m)*dp*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d-dp)*ts);
chi1=1i*wac(m)*I-a1;
chi2=1i*wac(m)*I-a2;
chi3=1i*wac(m)*I-a3;
eta1=chi1\ (I-beta1);
eta2=chi2\ (I-beta2);
eta3=chi3\ (I-beta3);
phasor1=I-beta3*Gamma2*beta2*Gamma1*beta1;
phasor2=beta3*Gamma2*beta2*Gamma1*eta1*b1;
phasor3=beta3*Gamma2*eta2*b2+eta3*b3;
phasor=phasor1\ (phasor2+phasor3);
beta1d=phi1d*exp(-1i*wac(m)*tstep);
eta1d=chi1\ (I-beta1d);
beta2d=phi2d*exp(-1i*wac(m)*tstep);
eta2d=chi2\ (I-beta2d);
beta3d=phi3d*exp(-1i*wac(m)*tstep);
eta3d=chi3\ (I-beta3d);
t(1)=0;
x=phasor;
y(1)=c1*x+e1;
n=2;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x+eta1d*b1*exp(1i*wac(m)*t(n));
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
end
phid=expm(a1*(d*ts-t(n-1)));

```

```

betad=phid*exp(-1i*wac(m)*(d*ts-t(n-1)));
etad=chi1/(I-betad);
t(n)=d*ts;
x=Gamma1*phid*x+Gamma1*etad*b1*exp(1i*wac(m)*d*ts);
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < (d+dp)*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x+eta2d*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
end
phidp=expm(a2*((d+dp)*ts-t(n-1)));
betadp=phidp*exp(-1i*wac(m)*((d+dp)*ts-t(n-1)));
etadp=chi2/(I-betadp);
t(n)=(d+dp)*ts;
x=Gamma2*phidp*x+Gamma2*etadp*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x+eta3d*b3*exp(1i*wac(m)*t(n));
y(n)=c3*x+e3*exp(1i*wac(m)*t(n));
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
betaend=phiend*exp(-1i*wac(m)*(ts-t(n-1)));
etaend=chi3/(I-betaend);
t(n)=ts;
x=phiend*x+etaend*b3*exp(1i*wac(m)*ts);
y(n)=c3*x+e3*exp(1i*wac(m)*ts);
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=(1/ts)*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% loadCPMDVM.m
% loads CPM OTB DVM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
vg=28;r=3;l1=130e-6;l2=1e-3;c1=60e-9;c2=50e-6;d=0.4;ts=1/100e3;
a1=[0 0 0 -1/l1;
0 0 -1/l2 1/l2;
0 1/c1 0 0;
1/c2 -1/c2 0 -1/(r*c2)];
a2=[0 0 0 -1/l1;
0 0 0 1/l2;
0 0 0 0;
1/c2 -1/c2 0 -1/(r*c2)];
a3=[0 0 -1/l1 -1/l1;
0 0 0 1/l2;
1/c1 0 0 0;
1/c2 -1/c2 0 -1/(r*c2)];
b1=[1/l1; 0; 0; 0];
b2=[1/l1; 0; 0; 0];
b3=[1/l1; 0; 0; 0];
c1=[0 0 0 1];
c2=[0 0 0 1];
c3=[0 0 0 1];
e1=0;
e2=0;
e3=0;
ht1=[0 0 1 0];
ht2=[1 -1 0 0];
m=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CPMDVMexact.m
% CPM OTB DVM exact results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadCPMDVM;
dv=fzero(@(x) dvm_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht1),d/2);
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,dv*ts);
[phi2,psi2]=c2d(a2,b2,(d-dv)*ts);
[phi3,psi3]=c2d(a3,b3,(1-d)*ts);
x2=(I-phi1*phi3*psi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;

```

```

xi2=(a2-a3)*x3+(b2-b3)*vg;
zeta1=(c1-c2)*x2+(e1-e2)*vg;
zeta2=(c2-c3)*x3+(e2-e3)*vg;
gamma1=a1*x2+b1*vg;
gamma2=a2*x3+b2*vg;
mu1=1/(ht1*gamma1);
mu2=1/(ht2*gamma2+m);
Gamma1=I-xi1*ht1*mu1;
Gamma2=I-xi2*ht2*mu2;

k=100;
f=logspace(2,5,k);
w=2*pi*f;
for n=1:k
beta1=exp(-1i*w(n)*dv*ts)*phi1;
beta2=exp(-1i*w(n)*(d-dv)*ts)*phi2;
beta3=exp(-1i*w(n)*(1-d)*ts)*phi3;
chi1=(1i*w(n)*I-a1);
chi2=(1i*w(n)*I-a2);
chi3=(1i*w(n)*I-a3);
eta1=chi1/(I-beta1);
eta2=chi2/(I-beta2);
eta3=chi3/(I-beta3);

co11=c3*eta3;
co12=c1*eta1*beta3;
co13=(c2*eta2-zeta2*mu2*ht2*beta2)*Gamma1*beta1*beta3;
co14=zeta1*mu1*ht1*beta1*beta3;
co1=co11+co12+co13-co14;

co2=(I-Gamma2*beta2*Gamma1*beta1*beta3)\xi2;
co(n)=(mu2/ts)*(co1*co2+zeta2);

rho1=c1*(chi1/(dv*ts*I-eta1))*b1+e1*dv*ts;
rho2=c2*(chi2/((d-dv)*ts*I-eta2))*b2+e2*(d-dv)*ts;
rho3=c3*(chi3/((1-d)*ts*I-eta3))*b3+e3*(1-d)*ts;

io11=c3*eta3;
io12=c1*eta1*beta3;
io13=(c2*eta2-zeta2*mu2*ht2*beta2)*Gamma1*beta1*beta3;
io14=zeta1*mu1*ht1*beta1*beta3;
io1=io11+io12+io13-io14;

io21=I-Gamma2*beta2*Gamma1*beta1*beta3;

```

```

io22=Gamma2*beta2*Gamma1*beta1*eta3*b3;
io23=Gamma2*beta2*Gamma1*eta1*b1;
io24=Gamma2*eta2*b2;
io2=io21\ (io22+io23+io24);

io311=c1*eta1*eta3*b3;
io312=(c1*eta2-zeta2*mu2*ht2*beta2)*Gamma1*beta1*eta3*b3;
io313=zeta1*mu1*ht1*beta1*eta3*b3;
io31=io311+io312-io313;
io32=((c2*eta2-zeta2*mu2*ht2*beta2)*Gamma1-zeta1*mu1*ht1)*eta1*b1;
io33=zeta2*mu2*ht2*eta2*b2;
io3=io31+io32-io33;

io(n)=(1/ts)*(io1*io2+io3+rho1+rho2+rho3);
end

comag=20*log10(abs(co));
cophase=180/pi*unwrap(angle(co));
iomag=20*log10(abs(io));
iophase=180/pi*unwrap(angle(io));

hold off
semilogx(f,comag)
pause
semilogx(f,cophase)
pause
semilogx(f,iomag)
pause
semilogx(f,iophase)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CPMDVMSimco.m
% CPM OTB DVM simulation control-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadCPMDVM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

dv=fzero(@(x)dvm_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht1),d/2);

```

```

I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,dv*ts);
[phi2,psi2]=c2d(a2,b2,(d-dv)*ts);
[phi3,psi3]=c2d(a3,b3,(1-d)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
gamma1=a1*x2+b1*vg;
gamma2=a2*x3+b2*vg;
mu2=1/(ht2*gamma2+m);
Gamma2=I-xi2*mu2*ht2;
mu1=1/(ht1*gamma1);
Gamma1=I-xi1*mu1*ht1;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*dv*ts);
beta2=phi2*exp(-1i*wac(m)*(d-dv)*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d)*ts);
phasor=(I-beta3*Gamma2*beta2*Gamma1*beta1)\beta3*xi2*mu2;

t(1)=0;
x=phasor;
y(1)=c1*x;
n=2;
while t(n-1)+tstep < dv*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x;
y(n)=c1*x;
n=n+1;
end
phid=expm(a1*(dv*ts-t(n-1)));
t(n)=dv*ts;
x=Gamma1*phid*x;
y(n)=c1*x;
n=n+1;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x;
y(n)=c2*x;

```

```

n=n+1;
end
phidp=expm(a2*(d*ts-t(n-1)));
t(n)=d*ts;
x=Gamma2*phidp*x+xi2*mu2*exp(1i*wac(m)*d*ts);
y(n)=c2*x;
n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x;
y(n)=c3*x;
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
t(n)=ts;
x=phiend*x;
y(n)=c3*x;
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=1/ts*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CPMDVMSimio.m
% CPM OTB DVM simulation input-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadCPMDVM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac = 2*pi*fac;

```



```

dv=fzero(@(x)dvm_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht1),d/2);
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,dv*ts);
[phi2,psi2]=c2d(a2,b2,(d-dv)*ts);
[phi3,psi3]=c2d(a3,b3,(1-d)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
gamma1=a1*x2+b1*vg;
gamma2=a2*x3+b2*vg;
mu2=1/(ht2*gamma2+m);
Gamma2=I-xi2*mu2*ht2;
mu1=1/(ht1*gamma1);
Gamma1=I-xi1*mu1*ht1;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*dv*ts);
beta2=phi2*exp(-1i*wac(m)*(d-dv)*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d)*ts);
chi1=1i*wac(m)*I-a1;
chi2=1i*wac(m)*I-a2;
chi3=1i*wac(m)*I-a3;
eta1=chi1\ (I-beta1);
eta2=chi2\ (I-beta2);
eta3=chi3\ (I-beta3);
phasor1=I-beta3*Gamma2*beta2*Gamma1*beta1;
phasor2=beta3*Gamma2*beta2*Gamma1*eta1*b1;
phasor3=beta3*Gamma2*eta2*b2+eta3*b3;
phasor=phasor1\ (phasor2+phasor3);
beta1d=phi1d*exp(-1i*wac(m)*tstep);
eta1d=chi1\ (I-beta1d);
beta2d=phi2d*exp(-1i*wac(m)*tstep);
eta2d=chi2\ (I-beta2d);
beta3d=phi3d*exp(-1i*wac(m)*tstep);
eta3d=chi3\ (I-beta3d);

t(1)=0;
x=phasor;
y(1)=c1*x+e1;

```

```

n=2;
while t(n-1)+tstep < dv*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x+eta1d*b1*exp(1i*wac(m)*t(n));
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
end
phid=expm(a1*(dv*ts-t(n-1)));
betad=phid*exp(-1i*wac(m)*(dv*ts-t(n-1)));
etad=chi1/(I-betad);
t(n)=dv*ts;
x=Gamma1*phid*x+Gamma1*etad*b1*exp(1i*wac(m)*dv*ts);
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x+eta2d*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
end
phidp=expm(a2*(d*ts-t(n-1)));
betadp=phidp*exp(-1i*wac(m)*(d*ts-t(n-1)));
etadp=chi2/(I-betadp);
t(n)=d*ts;
x=Gamma2*phidp*x+Gamma2*etadp*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x+eta3d*b3*exp(1i*wac(m)*t(n));
y(n)=c3*x+e3*exp(1i*wac(m)*t(n));
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
betaend=phiend*exp(-1i*wac(m)*(ts-t(n-1)));
etaend=chi3/(I-betaend);
t(n)=ts;
x=phiend*x+etaend*b3*exp(1i*wac(m)*ts);
y(n)=c3*x+e3*exp(1i*wac(m)*ts);
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=(1/ts)*trapz(t,y);

```

```

end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% loadV2CCM.m
% loads V2 boost CCM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lf=58e-6;cf=5.5e-6;r=18.6;vg=15;d=0.25;ts=1/100e3;
a1=[0 0;0 -1/(r*cf)];
a2=[0 -1/lf;1/cf -1/(r*cf)];
b1=[1/lf; 0];
b2=[1/lf; 0];
c1=[0 1];
c2=[0 1];
e1=0;
e2=0;
I=eye(size(a1));
ht=[0 1];
m=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V2CCMexact.m
% V2 boost CCM exact results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadV2CCM;
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,(1-d)*ts);
x2=(I-phi1*phi2)\(phi1*psi2+psi1)*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
zeta1=(c1-c2)*x2+(e1-e2)*vg;
gamma1=a1*x2+b1*vg;
mu1=1/(ht*gamma1+m);
Gamma1=I-xi1*ht*mu1;

k=300;
f=logspace(2,5,k);
w=2*pi*f;
for n=1:k

```

```

beta1=exp(-1i*w(n)*d*ts)*phi1;
beta2=exp(-1i*w(n)*(1-d)*ts)*phi2;
chi1=(1i*w(n)*I-a1);
chi2=(1i*w(n)*I-a2);
eta1=chi1/(I-beta1);
eta2=chi2/(I-beta2);

co1=c1*eta1*beta2+c2*eta2-zeta1*ht*mu1*beta1*beta2;
co2=(I-Gamma1*beta1*beta2)/xi1;
co(n)=(mu1/ts)*(co1*co2+zeta1);

rho1=c1*(chi1/(d*ts*I-eta1))*b1+e1*d*ts;
rho2=c2*(chi2/((1-d)*ts*I-eta2))*b2+e2*(1-d)*ts;

io1=c1*eta1*beta2+c2*eta2-zeta1*mu1*ht*beta1*beta2;
io2=I-Gamma1*beta1*beta2;
io3=Gamma1*beta1*eta2*b2+Gamma1*eta1*b1;
io4=c1*eta1*eta2*b2-zeta1*mu1*ht*(eta1*b1+beta1*eta2*b2);
io(n)=(1/ts)*(io1*(io2/io3)+io4+rho1+rho2);
end

comag=20*log10(abs(co));
cophase=180/pi*unwrap(angle(co));
iomag=20*log10(abs(io));
iophase=180/pi*unwrap(angle(io));

hold off
semilogx(f,comag)
pause
semilogx(f,cophase)
pause
semilogx(f,iomag)
pause
semilogx(f,iophase)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V2CCMsimco.m
% V2 boost CCM simulation control-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadV2CCM;
lowfreq=100;
highfreq=100000;
Ms=100;

```

```

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,(1-d)*ts);
x2=(I-phi1*phi2)\(phi1*psi2+psi1)*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
zeta1=(c1-c2)*x2+(e1-e2)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
mu1=1/(ht*(a1*x2+b1*vg));
Gamma1=I-xi1*ht*mu1;

for m=1:length(fac)
    beta1=phi1*exp(-1i*wac(m)*d*ts);
    beta2=phi2*exp(-1i*wac(m)*(1-d)*ts);
    phasor=(I-beta2*Gamma1*beta1)\beta2*xi1*mu1;

    t(1)=0;
    x=phasor;
    y(1)=c1*x;
    n=2;
    while t(n-1)+tstep < d*ts;
        t(n)=t(n-1)+tstep;
        x=phi1d*x;
        y(n)=c1*x;
        n=n+1;
    end
    phid=expm(a1*(d*ts-t(n-1)));
    t(n)=d*ts;
    x=Gamma1*phid*x+xi1*mu1*exp(1i*wac(m)*d*ts);
    y(n)=c1*x;
    n=n+1;
    while t(n-1)+tstep < ts;
        t(n)=t(n-1)+tstep;
        x=phi2d*x;
        y(n)=c2*x;
        n=n+1;
    end
    phiend=expm(a2*(ts-t(n-1)));
    t(n)=ts;

```

```

x=phiend*x;
y(n)=c2*x;
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=1/ts*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V2CCMsimio.m
% V2 boost CCM simulation input-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadV2CCM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac = 2*pi*fac;

I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,(1-d)*ts);
x2=(I-phi1*phi2)\(phi1*psi2+psi1)*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
zeta1=(c1-c2)*x2+(e1-e2)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
mu1=1/(ht*(a1*x2+b1*vg));
Gamma1=I-xi1*ht*mu1;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*d*ts);
beta2=phi2*exp(-1i*wac(m)*(1-d)*ts);

```

```

chi1=1i*wac(m)*I-a1;
chi2=1i*wac(m)*I-a2;
eta1=chi1/(I-beta1);
eta2=chi2/(I-beta2);
phasor=(I-beta2*Gamma1*beta1)/(beta2*Gamma1*eta1*b1+eta2*b2);

beta1d=phi1d*exp(-1i*wac(m)*tstep);
eta1d=chi1/(I-beta1d);
beta2d=phi2d*exp(-1i*wac(m)*tstep);
eta2d=chi2/(I-beta2d);

t(1)=0;
x=phasor;
y(1)=c1*x+e1;
n=2;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x+eta1d*b1*exp(1i*wac(m)*t(n));
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
end
phid=expm(a1*(d*ts-t(n-1)));
betad=phid*exp(-1i*wac(m)*(d*ts-t(n-1)));
etad=chi1/(I-betad);
t(n)=d*ts;
x=Gamma1*phid*x+Gamma1*etad*b1*exp(1i*wac(m)*d*ts);
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < ts;
t(n)=t(n-1)+tstep;
x=phi2d*x+eta2d*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
end
phiend=expm(a2*(ts-t(n-1)));
betaend=phiend*exp(-1i*wac(m)*(ts-t(n-1)));
etaend=chi2/(I-betaend);
t(n)=ts;
x=phiend*x+etaend*b2*exp(1i*wac(m)*ts);
y(n)=c2*x+e2*exp(1i*wac(m)*ts);
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end

```

```

z(m)=(1/ts)*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% loadV2DCM.m
% loads V2 boost DCM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lf=58e-6;cf=5.5e-6;r=150;rs=0.5;vg=15;d=0.25;ts=1/100e3;
a1=[-rs/lf 0;0 -1/(r*cf)];
a2=[0 -1/lf;1/cf -1/(r*cf)];
a3=[0 0;0 -1/(r*cf)];
b1=[1/lf; 0];
b2=[1/lf; 0];
b3=[0; 0];
c1=[0 1];
c2=[0 1];
c3=[0 1];
e1=0;
e2=0;
e3=0;
ht1=[0 1];
ht2=[1 0];
m=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V2DCMexact.m
$ V2 boost DCM exact results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadV2DCM;
dp=fzero(@(x)dp_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht2),(1-d)/2);
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,dp*ts);
[phi3,psi3]=c2d(a3,b3,(1-d-dp)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;

```



```

xi2=(a2-a3)*x3+(b2-b3)*vg;
zeta1=(c1-c2)*x2+(e1-e2)*vg;
zeta2=(c2-c3)*x3+(e2-e3)*vg;
m=0;
gamma1=a1*x2+b1*vg;
gamma2=a2*x3+b2*vg;
mu1=1/(ht1*gamma1+m);
mu2=1/(ht2*gamma2);
Gamma1=I-xi1*ht1*mu1;
Gamma2=I-xi2*ht2*mu2;

k=100;
f=logspace(2,5,k);
w=2*pi*f;
for n=1:k
beta1=exp(-1i*w(n)*d*ts)*phi1;
beta2=exp(-1i*w(n)*dp*ts)*phi2;
beta3=exp(-1i*w(n)*(1-d-dp)*ts)*phi3;
chi1=(1i*w(n)*I-a1);
chi2=(1i*w(n)*I-a2);
chi3=(1i*w(n)*I-a3);
eta1=chi1/(I-beta1);
eta2=chi2/(I-beta2);
eta3=chi3/(I-beta3);

co1=(c1*eta1-zeta1*mu1*ht1*beta1)*beta3*Gamma2*beta2;
co2=c2*eta2-zeta2*mu2*ht2*beta2+c3*eta3*Gamma2*beta2;
co3=I-Gamma1*beta1*beta3*Gamma2*beta2;
co(n)=(mu1/ts)*((co1+co2)*(co3\xi1)+zeta1);

rho1=c1*(chi1/(d*ts*I-eta1))*b1+e1*d*ts;
rho2=c2*(chi2/(dp*ts*I-eta2))*b2+e2*dp*ts;
rho3=c3*(chi3/((1-d-dp)*ts*I-eta3))*b3+e3*(1-d-dp)*ts;

io1=c2*eta2-zeta2*mu2*ht2*beta2+c3*eta3*Gamma2*beta2;
io2=(c1*eta1-zeta1*mu1*ht1*beta1)*beta3*Gamma2*beta2;
io3=I-Gamma1*beta1*beta3*Gamma2*beta2;

io41=Gamma1*beta1*beta3*Gamma2*eta2*b2;
io42=Gamma1*beta1*eta3*b3+Gamma1*eta1*b1;
io4=io41+io42;

io51=c3*eta3*Gamma2*eta2*b2;
io52=c1*eta1*beta3*Gamma2*eta2*b2;

```

```

io53=c1*eta1*eta3*b3;
io5=io51+io52+io53;

io61=zeta2*mu2*ht2*eta2*b2;
io62=zeta1*mu1*ht1*beta1*beta3*Gamma2*eta2*b2;
io6=io61+io62;

io7=zeta1*mu1*ht1*beta1*eta3*b3+zeta1*mu1*ht1*eta1*b1;
io(n)=(1/ts)*((io1+io2)*(io3\io4)+io5-io6-io7+rho1+rho2+rho3);
end

comag=20*log10(abs(co));
cophase=180/pi*unwrap(angle(co));
iomag=20*log10(abs(io));
iophase=180/pi*unwrap(angle(io));

hold off
semilogx(f,comag);
pause
semilogx(f,cophase);
pause
semilogx(f,iomag);
pause
semilogx(f,iophase);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V2DCMsimco.m
% V2 boost DCM simulation control-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadV2DCM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

dp=fzero(@(x)dp_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht2),(1-d)/2);
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,dp*ts);
[phi3,psi3]=c2d(a3,b3,(1-d-dp)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;

```

```

x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
mu1=1/(ht1*(a1*x2+b1*vg)+m);
Gamma1=I-xi1*mu1*ht1;
mu2=1/(ht2*(a2*x3+b2*vg));
Gamma2=I-xi2*mu2*ht2;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*d*ts);
beta2=phi2*exp(-1i*wac(m)*dp*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d-dp)*ts);
phasor1=I-beta3*Gamma2*beta2*Gamma1*beta1;
phasor2=beta3*Gamma2*beta2*xi1*mu1;
phasor=phasor1\phasor2;

t(1)=0;
x=phasor;
y(1)=c1*x;
n=2;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x;
y(n)=c1*x;
n=n+1;
end
phid=expm(a1*(d*ts-t(n-1)));
t(n)=d*ts;
x=Gamma1*phid*x+xi1*mu1*exp(1i*wac(m)*d*ts);
y(n)=c1*x;
n=n+1;
while t(n-1)+tstep < (d+dp)*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x;
y(n)=c2*x;
n=n+1;
end
phidp=expm(a2*((d+dp)*ts-t(n-1)));
t(n)=(d+dp)*ts;
x=Gamma2*phidp*x;

```

```

y(n)=c2*x;
n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x;
y(n)=c3*x;
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
t(n)=ts;
x=phiend*x;
y(n)=c3*x;
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=1/ts*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));
semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V2DCMsimio.m
% V2 boost DCM simulation input-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadV2DCM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

dp=fzero(@(x)dp_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht2),(1-d)/2);
I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,d*ts);
[phi2,psi2]=c2d(a2,b2,dp*ts);
[phi3,psi3]=c2d(a3,b3,(1-d-dp)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;

```

```

x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
mu1=1/(ht1*(a1*x2+b1*vg)+m);
Gamma1=I-xi1*mu1*ht1;
mu2=1/(ht2*(a2*x3+b2*vg));
Gamma2=I-xi2*mu2*ht2;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*d*ts);
beta2=phi2*exp(-1i*wac(m)*dp*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d-dp)*ts);
chi1=1i*wac(m)*I-a1;
chi2=1i*wac(m)*I-a2;
chi3=1i*wac(m)*I-a3;
eta1=chi1\ (I-beta1);
eta2=chi2\ (I-beta2);
eta3=chi3\ (I-beta3);
phasor1=I-beta3*Gamma2*beta2*Gamma1*beta1;
phasor2=beta3*Gamma2*beta2*Gamma1*eta1*b1;
phasor3=beta3*Gamma2*eta2*b2+eta3*b3;
phasor=phasor1\ (phasor2+phasor3);

beta1d=phi1d*exp(-1i*wac(m)*tstep);
eta1d=chi1\ (I-beta1d);
beta2d=phi2d*exp(-1i*wac(m)*tstep);
eta2d=chi2\ (I-beta2d);
beta3d=phi3d*exp(-1i*wac(m)*tstep);
eta3d=chi3\ (I-beta3d);

t(1)=0;
x=phasor;
y(1)=c1*x+e1;
n=2;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x+eta1d*b1*exp(1i*wac(m)*t(n));
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
end

```

```

phid=expm(a1*(d*ts-t(n-1)));
betad=phid*exp(-1i*wac(m)*(d*ts-t(n-1)));
etad=chi1/(I-betad);
t(n)=d*ts;
x=Gamma1*phid*x+Gamma1*etad*b1*exp(1i*wac(m)*d*ts);
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < (d+dp)*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x+eta2d*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
end
phidp=expm(a2*((d+dp)*ts-t(n-1)));
betadp=phidp*exp(-1i*wac(m)*((d+dp)*ts-t(n-1)));
etadp=chi2/(I-betadp);
t(n)=(d+dp)*ts;
x=Gamma2*phidp*x+Gamma2*etadp*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x+eta3d*b3*exp(1i*wac(m)*t(n));
y(n)=c3*x+e3*exp(1i*wac(m)*t(n));
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
betaend=phiend*exp(-1i*wac(m)*(ts-t(n-1)));
etaend=chi3/(I-betaend);
t(n)=ts;
x=phiend*x+etaend*b3*exp(1i*wac(m)*ts);
y(n)=c3*x+e3*exp(1i*wac(m)*ts);
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=(1/ts)*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause

```

```

semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% loadV2DVM.m
% loads V2 OTB DVM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
vg=28;r=3;l1=130e-6;l2=1e-3;c1=60e-9;c2=50e-6;d=0.4;ts=1/100e3;
a1=[0 0 0 -1/l1;
0 0 -1/l2 1/l2;
0 1/c1 0 0;
1/c2 -1/c2 0 -1/(r*c2)];
a2=[0 0 0 -1/l1;
0 0 0 1/l2;
0 0 0 0;
1/c2 -1/c2 0 -1/(r*c2)];
a3=[0 0 -1/l1 -1/l1;
0 0 0 1/l2;
1/c1 0 0 0;
1/c2 -1/c2 0 -1/(r*c2)];
b1=[1/l1; 0; 0; 0];
b2=[1/l1; 0; 0; 0];
b3=[1/l1; 0; 0; 0];
c1=[0 0 0 1];
c2=[0 0 0 1];
c3=[0 0 0 1];
e1=0;
e2=0;
e3=0;
ht1=[0 0 1 0];
ht2=[0 0 0 1];
m=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V2DVMexact.m
% V2 OTB DVM exact results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadV2DVM;
I=eye(size(a1));
dv=fzero(@(x)dvm_interval( x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht1),d/2);
[phi1,psi1]=c2d(a1,b1,dv*ts);
[phi2,psi2]=c2d(a2,b2,(d-dv)*ts);
[phi3,psi3]=c2d(a3,b3,(1-d)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;

```

```

xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
zeta1=(c1-c2)*x2+(e1-e2)*vg;
zeta2=(c2-c3)*x3+(e2-e3)*vg;
gamma1=a1*x2+b1*vg;
gamma2=a2*x3+b2*vg;
mu1=1/(ht1*gamma1);
mu2=1/(ht2*gamma2);
Gamma1=I-xi1*ht1*mu1;
Gamma2=I-xi2*ht2*mu2;

k=100;
f=logspace(2,5,k);
w=2*pi*f;
for n=1:k
beta1=exp(-1i*w(n)*dv*ts)*phi1;
beta2=exp(-1i*w(n)*(d-dv)*ts)*phi2;
beta3=exp(-1i*w(n)*(1-d)*ts)*phi3;
chi1=(1i*w(n)*I-a1);
chi2=(1i*w(n)*I-a2);
chi3=(1i*w(n)*I-a3);
eta1=chi1/(I-beta1);
eta2=chi2/(I-beta2);
eta3=chi3/(I-beta3);

co11=c3*eta3;
co12=c1*eta1*beta3;
co13=(c2*eta2-zeta2*mu2*ht2*beta2)*Gamma1*beta1*beta3;
co14=zeta1*mu1*ht1*beta1*beta3;
co1=co11+co12+co13-co14;

co2=(I-Gamma2*beta2*Gamma1*beta1*beta3)\xi2;
co(n)=(mu2/ts)*(co1*co2+zeta2);

rho1=c1*(chi1/(dv*ts*I-eta1))*b1+e1*dv*ts;
rho2=c2*(chi2/((d-dv)*ts*I-eta2))*b2+e2*(d-dv)*ts;
rho3=c3*(chi3/((1-d)*ts*I-eta3))*b3+e3*(1-d)*ts;

io11=c3*eta3;
io12=c1*eta1*beta3;
io13=(c2*eta2-zeta2*mu2*ht2*beta2)*Gamma1*beta1*beta3;
io14=zeta1*mu1*ht1*beta1*beta3;
io1=io11+io12+io13-io14;

```



```

io2=(I-Gamma2*beta2*Gamma1*beta1*beta3)\Gamma2;
io3=beta2*(Gamma1*beta1*eta3*b3+eta1*b1)+eta2*b2;

io41=c1*eta1*eta3*b3;
io42=(c2*eta2-zeta2*mu2*ht2*beta2)*Gamma1*beta1*eta3*b3;
io43=zeta1*mu1*ht1*beta1*eta3*b3;
io4=io41+io42-io43;

io51=(c2*eta2-zeta1*mu1*ht1-zeta2*mu2*ht2*beta2)*eta1*b1;
io52=zeta2*mu2*ht2*beta2*eta2*b2;
io5=io51-io52;

io(n)=(1/ts)*(io1*io2*io3+io4+io5+rho1+rho2+rho3);
end

comag=20*log10(abs(co));
cophase=180/pi*unwrap(angle(co));
iomag=20*log10(abs(io));
iophase=180/pi*unwrap(angle(io));

hold off
semilogx(f,comag);
pause
semilogx(f,cophase);
pause
semilogx(f,iomag);
pause
semilogx(f,iophase);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V2DVMsimco.m
% V2 OTB DVM simulation control-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadV2DVM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

dv = fzero(@(x) dvm_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht1), d/2);
I=eye(size(a1));

```

```

[phi1,psi1]=c2d(a1,b1,dv*ts);
[phi2,psi2]=c2d(a2,b2,(d-dv)*ts);
[phi3,psi3]=c2d(a3,b3,(1-d)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
gamma1=a1*x2+b1*vg;
gamma2=a2*x3+b2*vg;
mu2=1/(ht2*gamma2);
Gamma2=I-xi2*mu2*ht2;
mu1=1/(ht1*gamma1);
Gamma1=I-xi1*mu1*ht1;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*dv*ts);
beta2=phi2*exp(-1i*wac(m)*(d-dv)*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d)*ts);
phasor=(I-beta3*Gamma2*beta2*Gamma1*beta1)\beta3*xi2*mu2;

t(1)=0;
x=phasor;
y(1)=c1*x;
n=2;
while t(n-1)+tstep < dv*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x;
y(n)=c1*x;
n=n+1;
end
phid=expm(a1*(dv*ts-t(n-1)));
t(n)=dv*ts;
x=Gamma1*phid*x;
y(n)=c1*x;
n=n+1;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x;
y(n)=c2*x;
n=n+1;

```

```

end
phidp=expm(a2*(d*ts-t(n-1)));
t(n)=d*ts;
x=Gamma2*phidp*x+xi2*mu2*exp(1i*wac(m)*d*ts);
y(n)=c2*x;
n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x;
y(n)=c3*x;
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
t(n)=ts;
x=phiend*x;
y(n)=c3*x;
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=1/ts*trapz(t,y);
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% V2DVMsimio.m
% V2 OTB DVM simulation input-to-output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loadV2DVM;
lowfreq=100;
highfreq=100000;
Ms=100;

k=300;
fac=logspace(log10(lowfreq),log10(highfreq),k);
wac=2*pi*fac;

dv=fzero(@(x)dvm_interval(x,a1,b1,a2,b2,a3,b3,ts,d,vg,ht1),d/2);

```

```

I=eye(size(a1));
[phi1,psi1]=c2d(a1,b1,dv*ts);
[phi2,psi2]=c2d(a2,b2,(d-dv)*ts);
[phi3,psi3]=c2d(a3,b3,(1-d)*ts);
x2=(I-phi1*phi3*phi2)\(phi1*phi3*psi2+phi1*psi3+psi1)*vg;
x3=phi2*x2+psi2*vg;
xi1=(a1-a2)*x2+(b1-b2)*vg;
xi2=(a2-a3)*x3+(b2-b3)*vg;
tstep=ts/Ms;
phi1d=expm(a1*tstep);
phi2d=expm(a2*tstep);
phi3d=expm(a3*tstep);
gamma1=a1*x2+b1*vg;
gamma2=a2*x3+b2*vg;
mu2=1/(ht2*gamma2);
Gamma2=I-xi2*mu2*ht2;
mu1=1/(ht1*gamma1);
Gamma1=I-xi1*mu1*ht1;

for m=1:length(fac)
beta1=phi1*exp(-1i*wac(m)*dv*ts);
beta2=phi2*exp(-1i*wac(m)*(d-dv)*ts);
beta3=phi3*exp(-1i*wac(m)*(1-d)*ts);
chi1=1i*wac(m)*I-a1;
chi2=1i*wac(m)*I-a2;
chi3=1i*wac(m)*I-a3;
eta1=chi1\ (I-beta1);
eta2=chi2\ (I-beta2);
eta3=chi3\ (I-beta3);
phasor1=I-beta3*Gamma2*beta2*Gamma1*beta1;
phasor2=beta3*Gamma2*beta2*Gamma1*eta1*b1;
phasor3=beta3*Gamma2*eta2*b2+eta3*b3;
phasor=phasor1\ (phasor2+phasor3);

beta1d=phi1d*exp(-1i*wac(m)*tstep);
eta1d=chi1\ (I-beta1d);
beta2d=phi2d*exp(-1i*wac(m)*tstep);
eta2d=chi2\ (I-beta2d);
beta3d=phi3d*exp(-1i*wac(m)*tstep);
eta3d=chi3\ (I-beta3d);

t(1)=0;
x=phasor;
y(1)=c1*x+e1;

```

```

n=2;
while t(n-1)+tstep < dv*ts;
t(n)=t(n-1)+tstep;
x=phi1d*x+eta1d*b1*exp(1i*wac(m)*t(n));
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
end
phid=expm(a1*(dv*ts-t(n-1)));
betad=phid*exp(-1i*wac(m)*(dv*ts-t(n-1)));
etad=chi1/(I-betad);
t(n)=dv*ts;
x=Gamma1*phid*x+Gamma1*etad*b1*exp(1i*wac(m)*dv*ts);
y(n)=c1*x+e1*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < d*ts;
t(n)=t(n-1)+tstep;
x=phi2d*x+eta2d*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
end
phidp=expm(a2*(d*ts-t(n-1)));
betadp=phidp*exp(-1i*wac(m)*(d*ts-t(n-1)));
etadp=chi2/(I-betadp);
t(n)=d*ts;
x=Gamma2*phidp*x+Gamma2*etadp*b2*exp(1i*wac(m)*t(n));
y(n)=c2*x+e2*exp(1i*wac(m)*t(n));
n=n+1;
while t(n-1)+tstep < ts
t(n)=t(n-1)+tstep;
x=phi3d*x+eta3d*b3*exp(1i*wac(m)*t(n));
y(n)=c3*x+e3*exp(1i*wac(m)*t(n));
n=n+1;
end
phiend=expm(a3*(ts-t(n-1)));
betaend=phiend*exp(-1i*wac(m)*(ts-t(n-1)));
etaend=chi3/(I-betaend);
t(n)=ts;
x=phiend*x+etaend*b3*exp(1i*wac(m)*ts);
y(n)=c3*x+e3*exp(1i*wac(m)*ts);
n=n+1;
for n=1:length(t);
y(n)=y(n)*exp(-1i*wac(m)*t(n));
end
z(m)=(1/ts)*trapz(t,y);

```

```
end

mg=20*log10(abs(z));
ph=180/pi*unwrap(angle(z));

semilogx(fac,mg);
pause
semilogx(fac,ph);
```